Universidad
Carlos III de Madrid

Universidad Carlos III de Madrid

Aerospace Engineering

Final Thesis

Force Decomposition of 3D Unsteady Aerodynamics for 2 Symmetric Flapping Wings

Miguel Segovia Mora

Supervisor: Cayetano Martínez Muriel

Abstract

Nowadays, the aerodynamic processes that are behind any flapping flight are not completely understood by the scientific community. Despite there are some numerical methods which calculate the forces in 2D and 3D, these processes fail when they give an insight about the actual processes happening through a flapping system. In addition, these tools require a high computational time and the costs they sometimes present are very high.

When flapping wings are studied, the method used is the force decomposition where the vortex structures acquire a decisive role for the development of these flights. For relevant studies, the 3D models seem to be the best option because the 2D ones do not generate results so similar to reality. For this reason, the optimum analysis for flapping wings is a 3D motion which is subjected to different kinematics and geometry conditions to see the evolution for each decomposed force.

During this work, a code which computes the decomposed forces using Chang's algorithm has been taken as a reference. This code could only compute the heaving motion for just one 3D wing. As a result of this project, it has been created a tool which is capable of computing the decomposed forces for 2 wings. In addition, though user inputs, the geometry and kinematic can be changed. The geometry by means of the Aspect Ratio and the kinematics through the radius of rotation. Once this is achieved, after the corresponding optimizations and validations, it has been studied the influence on results that a second wing could have on a first one. For this, it has been set a motion of Re=500 and reduced frequency k=1.

The conclusion is that when the fluid is incompressible, there are no instabilities through the spanwise direction. As a consequence of this, the influence that the second wing has on the first one is negligible.

Finally, different geometry and kinematic conditions have been presented with the same simulating conditions. Through this part, it has been analyzed the influence that the variation of these parameters has on the resultant forces. The conclusion for this analysis says that heaving motion produces a higher quantity of thrust and lift with respect the flapping one. This is due to the kinematics nature of each case. During the heaving process, the wing will move a higher quantity of air than the flapping one. Heaving motion produces constant vortex structures around the span, so it will produce a similar amount of forces over the spanwise. For the flapping case, the vortex structures decrease from the wing tip to the inboard tip, generating a smaller quantity of thrust and lift.

Resumen

A día de hoy, los procesos aerodinámicos que ocurren durante el vuelo de aleteo no son comprendidos en su totalidad. A pesar de que existen métodos numéricos que calculan las fuerzas en 2 y 3 dimensiones, estos procesos fallan a la hora de dar una visión de lo que realmente ocurre desde un punto de vista aerodinámico. Además estas herramientas requieren un tiempo computacional considerable y los costes suelen ser bastante elevados.

En el estudio del vuelo de aleteo, el método empleado es la descomposición de fuerzas donde las estructuras vorticiales adquieren un papel decisivo en el desarrollo de estos vuelos. Para obtener resultados relevantes, los modelos tridimensionales son la mejor opción ya que aquellos en 2D no generan resultados tan similares a la realidad. Por ello, el análisis más óptimo que se puede hacer de vuelo en aleteo es un modelo en 3 dimensiones al cual se le someten diferentes tipos de movimientos y geometrías para comprobar la evolución de cada una de las fuerzas que intervienen en dicho movimiento.

Este trabajo, se basa en un código que obtenía la descomposición de fuerzas utilizando el algoritmo de Chang [22]. Este código sólo admitía movimiento agitado para un único ala. El resultado de este estudio es una herramienta que permite calcular la descomposición de fuerzas para 2 alas simétricas a las que se puede someter diferentes condiciones de geometría y cinemática. La parte geométrica en el sentido del Aspect Ratio; y la parte cinemática cambiando el radio de rotación para obtener diferentes movimientos de aleteo. Una vez conseguido, tras las consiguientes optimizaciones y validaciones, se ha estudiado la influencia que una segunda ala puede tener en los resultados de las fuerzas calculadas en la primera. Para esto se ha establecido un Re=500 y una frecuencia reducida de k=1. La conclusión a la que se llega es que cuando el flujo es incompresible, para los campos vorticiales no hay inestabilidades a lo largo de la dirección de enveradura de las alas. Esto hace que la influencia de la segunda ala sobre la primera pueda considerarse despreciable.

Finalmente, diferentes condiciones de geometría y cinemática se han establecido para estudiar la repercusión que estos dos parámetros pueden tener en la descomposición de fuerzas. La conclusión de este análisis dice que el movimiento agitado va a producir una mayor cantidad de empuje y sustentación respecto al vuelo en aleteo. Esto es debido a la naturaleza cinemática de cada movimiento: durante el proceso de agitado, el ala mueve mas masa de aire que el aleteo, por lo cual mayores fuerzas vorticiales se generarán en el cuerpo que más masa mueva.

List of Figures

List of Tables

Contents

1 Introduction

1.1 Motivation

The flights that some insects and small animals have during any conventional day have presented certain interest to human knowledge due to the complexity and uncertainty of the mechanism of this motion. This kind of flights rarely have been studied deeply in a theoretical, numerical or experimental point of view through aeronautical history. Some researches in the past have demonstrated the lower aerodynamic efficiency for flapping motion in high dimensions vehicles (i.e. high Reynolds Number). This is why the aeronautical industry has centered its attention in developing the fixed wing sector due to high demand of models which operate in a high Reynolds number range. [1]

Last decades, the increment of smaller vehicles with the appearance of nanotechnology has helped to stimulate the interest in Micro Air Vehicles (MAVs). Current uses for these vehicles can go from leisure, surveillance and security activities to search, mining and agriculture. In addition, it is known that the MAVs industry will even increase during next decades [2]. For this reason, it is needed to better understand what is behind this unsteady aerodynamics of flapping wings in the low Reynolds number range. Probably, one of the best ways is to look how the nature has adapted for millions of years to the flapping motions to ensure complex maneuvers as the ones performed, for instance, by the hummingbirds, bees or mosquito's [3]. After this, the knowledge taken from researches and studies will be applied in the development of these vehicles which are booming.

From an aerodynamic point of view, for designing a flapping system, it is not only needed the correct geometry design with the most sophisticated NACA profile, as usually happens with fixed wing models. It will be also taken into account cyclic parameters which will be decisive for the efficiency of the flight. For a given amplitude and frequency, another key point for the analysis will be the characterization of the vortex flow structures responsible of the thrust efficiency. Furthermore, these complex aerodynamic structures which appear in these models deal with difficult formulas which can be only solved by using sophisticated software. These Numerical analysis processes have been used to a considered extent in this field, having developed important advances during last years. These computing technology advances have led to a reduce in costs and time obtaining results which match so close to the reality.

Also the designed device must be able to react in a quick way to different changes in the environment. Here comes the maneuvility of the device, being the frequency crucial for changing the lift and thrust in the fastest way possible without losing the flight efficiency. This harmonization is done by studying separately the movements that produce these flights: heaving, pitching or flapping and seeing the consequences they have on each maneuver: hovering, vertical or forward flight.

From other similar motions as the one that some aquatic animals have, it has been obtained that the net thrust is related to the reversed von Kármán vortex [4]. Furthermore,

in most of the mechanisms studied for the flapping motion, the leading edge vortex (LEV) will be the main lift contributor for this analysis. It is also known that the cyclic motion generates a transient lift due to a LEV and it avoids entering in the dynamic stall of the system. [5]. In addition, it has been demonstrated that the largest thrust efficiency is achieved from a perfect interaction between the LEV and the trailing edge vortex (TEV) producing two vortices per stroke [6].

Hence, the aim of this study is to advance in the understanding of the flapping flight delivering support for future aerodynamic studies. Through this project two identical 3D wings situated making a mirror symmetry will be presented. Also the consequences that different kinematics and geometries have will be analyzed. For this, a decomposition of the aerodynamic forces which are in charge of producing a forward motion will be done.

1.2 State of the art

The nature motion of the flapping wing and the question for why some animals can fly by moving parts of their body, is a problem that the humans have been proposing during their existence. It was not until last decades that a strong interest on finding what is behind this flight physics has arised. The first official introduction about flapping wings is written by W. Shyy, H. Aono, C. K. Kang, and H. Liu in this document [7].

The improvements in photography and in particle tracing have permitted science to analyze through visual inspection the behavior of the insects and birds, allowing this step to reproduce closely the flapping motion. [8].

From here, numerous experimental studies have been performed for analysing as close as possible the real flight of animals as calliope hummingbird [9]. The results affirms that for the downstroke and upstroke of each wingbeat cycle, the thrust produced is such enough to overcome the drag of the animal and this thrust production pays back a negative lift induced during upstroke. Attending to the actual power required to develop these flights, there are experiments about the locomotion that birds need to achieve flapping motion, as the one carried out by Dececchi, T. Alexande in [10]. The results suggest that the wing expansion and elongation in flying animals cannot be intrinsically linked to locomotive adaptations, and there is a separation where it should be understood the origin of powered flight and avian evolution.

On the field of simulations, one of the optimum way of performing them is by Direct Numerical Simulation (DNS) methods as MArtín Alcántara and Moriche in [11] and [6]. Multiple kind of flights have been analyzed with DNS like vertical flight [12], hovering [13] and forward flight [14]. Lately, the question of how the short wings of mosquitoes can produce enough aerodynamic forces for sustaining such a long body, has led to multiple

conclusions included in [15]. The aeronautical community has tried to simplify as much as possible the problem statement changing from 3D to 2D case. Through this last alternative, it has been modelled analysis for heaving motion with an infinite Aspect Ratio as in[16] or [17].

On the other hand, it has been observed that during a conventional flight of an animal, pitching contribution also appears. This maximizes the flight efficiency of the creature. Some researches have been performed to study this efficiency as in [18]. With enough theoretical and empirical information, recent MAVs projects have been developed. This joined with the advances in Artificial Intelligence (AI), have permitted to build MAVs able to develop autonomous flight as the DelFLY Explorer with tail control [19]. Parallel, tailless MAVs have been created as Robobee by Harvard University which can adhere to the surface by a wired flight [20]. Nowadays, one of the most sophisticated tailless MAV is the KUBeetle, designed in Konkun university. It is characterized by the capability of autonomously doing all the flight physics maneuvers just by using an attitude controller [21].

For the statement of the decomposition problem, it is needed to present algorithms that simplify the conventional equations that govern the classic aerodynamic. The best example of this, is the force decomposition algorithm developed by Chang [22] in 1992 , taken as reference by multiple authors through next researches [14]. With the help of Chang's, Moriche in [6] derived an expression for the decomposition forces which will be used in other studies as Paulete's in [23].

Currently, the community is investigating models which are able to have as much degrees of freedom and geometric shapes as the real ones. This task is still in development and different prototypes should be analyzed before starting with the assembly of a complex real one. As it was expected, most of the simulations present some discrepancies with a real one because of the high unsteady aerodynamic present due to the high interaction that the vortex structures have during any flapping flight.

1.3 Objectives

The main objective of this thesis is to develop a tool which is capable of obtaining the decomposition of the forces for any forward flapping motion of two symmetric wings using the formulation proposed by Chang.

Previously, there was a code implemented by researches from Universidad Carlos III de Madrid where the only input available was 1 wing in Heaving motion. Through this code the decomposition forces: vorticity within the flow, surface vorticity and added mass terms were obtained. For doing this, the code needed the DNS data obtained by the CFD group

of the Bioengineering and Aerospace Engineering department of the Universidad Carlos III de Madrid. During this project, it will proposed an alternative code which is capable of admitting a pair of symmetric wings where the geometry can be varied and their movement can acquire a character of flapping or heaving depending on the input from the user. This tool will be also capable of calculating the decomposition of forces for these models using a method similar to the previous one introduced. The new proposed models will also simulate a forward flapping motion under unsteady aerodynamics.

Due to the complexity that this new code will have, some optimization processes should be performed to avoid memory performance overhead and excessive computational time. All these changes must be implemented trying to obtain the best accuracy results. After all the code is set, it should be validated that the tool works properly for any 3D body.

Once the tool is perfectly validated and prepared, it will be demonstrated the influence that a second wing has on the overall force decomposition with respect to results from the previous code with one body. Furthermore, different configurations and kinematics will be presented to study how the aerodynamic contributions to the total force change for each case. There will be 5 cases that will vary the kinematics from heaving to flapping motion and 1 case where the geometry parameters are changed.

With this, it will be attempted to advance as much as possible the actual knowledge in the MUAVs world. It will be deepened in the study of the terms that intervene in the creation of lift and drag. For this, it must be understood which decomposed forces are decisive to ensure the forward flapping flight when the geometry and kinematics parameters are changed.

2 Methodology

This Section will present the contents of the code implemented to calculate the decomposition of forces. First, an introduction to the problem will be done in Section 2.1 where the body and kinematics applied to both wings will be explained. Here it will treated the assembly of the body and input requirements for the code to generate the desired 3D wing.

After that, the Force Decomposition analysis will be performed during Section 2.2 where the expression of the total force used by the code will be figured out.

Finally a deep analysis of the code will be done from Section 2.3 to 2.7 where the assumptions, validations, refinements, and optimization's will be explained and reasoned. This last part will deal with the methodology and the numerical computations that each software will perform during its interventions along the whole code.

2.1 Problem statement

The problem proposed during this project is to compute the force decomposition applied to the 3D unsteady aerodynamic for two symmetric flapping wings with a constant inflow velocity U_∞. This will simulate the forward flapping flight of a body at this velocity with a Reynold Number Re=500. The body will be situated in a control volume in which the aerodynamic information will be obtained from a DNS analysis.

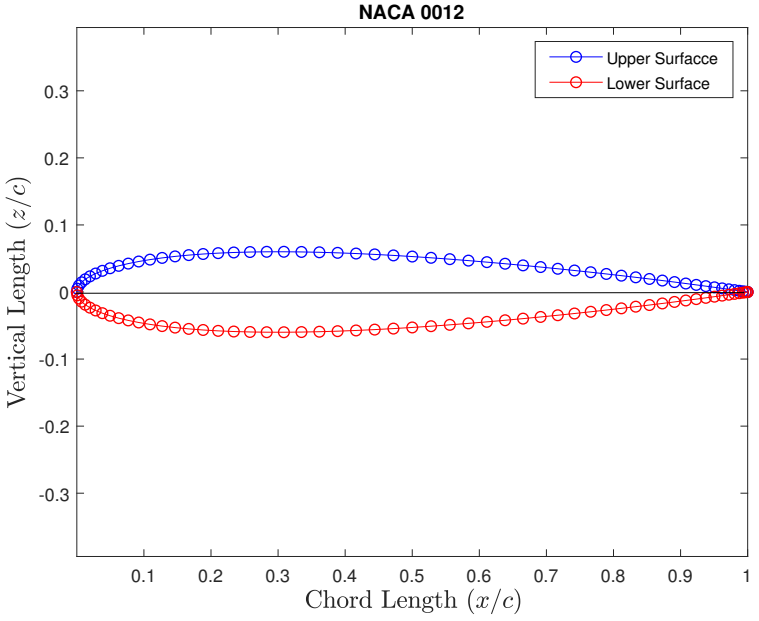


Figure 1: NACA0012 profile with 56 points per chord. [25]

Looking at the geometry, a 3D wing is considered, so the aspect ratio will be finite and defined as $AR = b/c$ where b is the wingspan and c is the chord. During almost all models, AR has been considered to be 2. But at the end of the thesis one case will be presented with $AR = 4$ to see how the geometry affects the results. The wing will have a rectangular planform with rounded edges on both tips and no geometric twist.

A symmetric airfoil profile has been chosen during the study. The profile is given by a NACA 0012, which is plotted in Figure 1. For this plot it has been taken into account 56 points per chord. This corresponds with the grid refinement of the control volume taken from the DNS data downloaded from TUCAN[1]. This DNS grid refinement analysis was performed by Alejandro Gonzalo Grande [14].

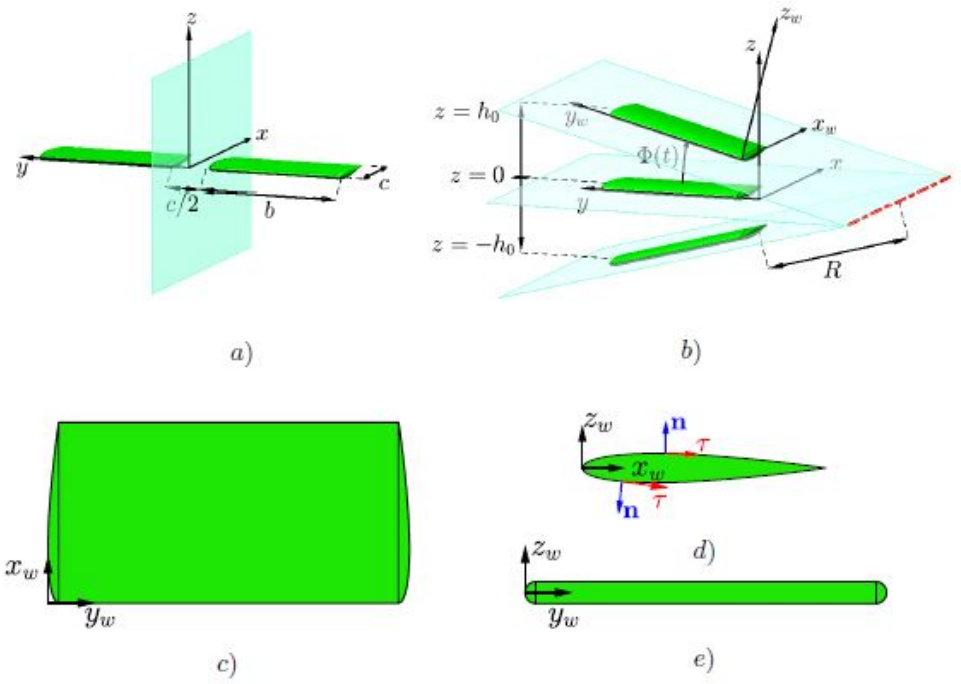


Figure 2: Wing configurations and Reference Frames attached to the wings [14]

From a kinematic point of view, it will be analyzed the transition between flapping to heaving for a forward flight. This transition is achieved by imposing an axis of rotation parallel to the free stream velocity and located a distance from the inboard tip of the wing.

The flapping angle will be defined such that it intrinsically depends on the AR and the radius of rotation (R). So heaving or flapping motion can be imposed by changing these two parameters. This angle, defined as Φ will follow a sinusoidal law with time:

[1]TUCAN will be the software used for the Direct Numerical Simulation (DNS).

$$\Phi = \Phi_m + \Phi_0 cos(kt + \varphi) \tag{1}$$

where Φ_m is the mean oscillating flapping angle, φ the phase shift angle and k is the reduced frequency defined as:

$$k = \frac{2\pi f c}{U_\infty} \tag{2}$$

where f is the flapping frequency of the motion related through $f = 1/T$ with the period T. The parameter Φ_0 is defined as:

$$\Phi_0 = arcsin\Big(\frac{h_0/c}{R/c + AR}\Big) \tag{3}$$

Where h_0 is the flapping amplitude as observed in Figure 2.

During all the cases, it has been constraint $h_0/c = 1$. This forces the outboard tip velocity to be the same for all cases of R and AR. This position corresponds with the middle of the downstroke and upstroke, thus $\frac{t}{T} = 0.25$ and $\frac{t}{T} = 0.75$ respectively. So imposing this condition, obligates to have the same kinematic conditions at the outboard tip for all the cases.

It could be also studied the influence of φ and the Φ_m on the forces, but due to the limited time for the project, it can be purposed as a future investigation.

Taking as reference Figure 2a, both wings will be situated a distance $a = c/2$ from the plane of symmetry Y=0 (measured from the inboard of both wings). The pivoting line situated in the y-axis, could be observed with a dashed red line in Figure 2b. Using Equations 1 and 3 and fixing AR=2, when $R = 0$, the motion is considered as pure flapping period being $\Phi_0 = \pi/6$. If $R \to \infty$ it results in a pure heaving motion where $\Phi_0 = 0$. It should be pointed that the flapping motion will be always symmetric with respect the Plane Y=0 so the value of R will be the same on both wings.

An inertial reference x, y, z system is considered which moves at the same direction and opposite sense of the inflow coming with speed U_∞. As observed in Figure 2a, it is attached in the middle of both wings, and at the leading edge of the airfoil profile. The direction through the span of the wing when $\Phi = 0$ corresponds with y direction and the vertical direction will be z. In this reference frame, the forward motion results in an incoming free stream along the positive direction of x axis.

Furthermore two additional non-inertial reference frame x_w, y_w, z_w has been taken into consideration: one attached to the inboard wing tip of the right wing, the other one attached to the left inboard wing tip. For both of them, the x_w corresponds with the chordwise direction, z_w is the direction perpendicular to the plane formed by the surface of the body and y_w is the spanwise direction along the wing. These reference frames moves with the

body following the sinusoidal motion imposed by Equation 1 in the X$_=$0*plane*.

In Figures 2c, 2d and 2e the three perspectives are represented where it can be observed where are the non-inertial references frames attached through each view. Furthermore, it can be seen the shear and the normal vector direction on the surface of the wing.

It should be pointed that in the code implemented, the reference frames taken change. This was because BEMLIB and *gmsh* use different coordinate system than the data taken from TUCAN. During this paper thesis it has been used the reference frame used by TUCAN and by Alejandro Gonzalo Grande in [14]. The Z-direction in the code corresponds with the negative Y-direction in Figure 2 and the Y-direction in the code will corresponds with the positive Z-direction in Figure 2.

2.2 Decomposition of forces

Before starting with the code, it should be introduced the decomposition of forces algorithm. With this it can be understood why certain parameters are computed with the code.

The total aerodynamic force $\vec{F}$ has been decomposed through Chang's algorithm [22] with the support of the method used by Moriche in [6]. The fluid to be studied has been considered incompressible so that the density is set to be constant in all the fluid domain. This makes the continuity equation have the following expression:

$$\nabla \cdot \vec{u} = 0 \tag{4}$$

From the Navier-Stokes momentum equation:

$$\frac{\partial \vec{u}}{\partial t} + (\vec{u} \cdot \nabla)\vec{u} = -\nabla p + \nu \nabla^2 \vec{u}, \tag{5}$$

it appears the pressure term which has been previously studied that generates errors for numerical computations when the flow is incompressible [32]. So the aim of the Chang's algorithm is to eliminate the contribution of the pressure to the total force .

Knowing that $(\vec{u} \cdot \nabla)\vec{u} = \nabla \frac{1}{2}\vec{u}^2 - \vec{u} \times \vec{w}$ and $\nu \nabla^2 \vec{u} = \nabla \times \vec{w}$, the next step is to multiply Equation 5 by $\frac{\nabla \phi_i}{U_\infty}$ and perform the volume integral for the control volume in both sides of the resulting identity. During these models, the incoming velocity to the wing U_∞ is considered constant with time.

The term ϕ_i is the potential flow and the subindex i is each of the Cartesian directions.

So the resulting formula is:

$$-\int_V \frac{\nabla\phi_i}{U_\infty}\cdot\nabla p dV = \int_V \frac{\nabla\phi_i}{U_\infty}\frac{\partial\vec{u}}{\partial t}dV + \frac{1}{2}\int_V \frac{\nabla\phi_i}{U_\infty}\cdot\nabla(|\vec{u}|^2)dV - \int_V \frac{\nabla\phi_i}{U_\infty}\cdot(\vec{u}\times\vec{w})dV + \nu\int_V \frac{\nabla\phi_i}{U_\infty}\cdot(\nabla\times\vec{w})dV \tag{6}$$

For avoiding extra computations that the control volume equations can deal, some of the volume terms can be rewrite in term of the surface integral speeding up the numerical calculations. So the equation:

$$\int_V \nabla\cdot\vec{F}dV = \int_S \vec{n}\cdot\vec{F}dA \tag{7}$$

with the hand of the following two identities $\nabla\cdot(f\vec{A}) = f\nabla\cdot\vec{A} + \vec{A}\cdot f$ and $\nabla\cdot(\vec{A}\times\vec{B}) = \vec{B}\cdot\nabla\times\vec{A} - \vec{A}\cdot\nabla\times\vec{B}$, will help the simplification of the left term and the first two terms of the right in the Equation 6. The vector $\vec{n}$ is the unitary normal vector to the surface of the wing pointing always toward the fluid.

So the simplifications are:

$$\frac{\nabla\phi_i}{U_\infty}\cdot\nabla p = \nabla\cdot(p\frac{\phi_i}{U_\infty}) - \cancel{p\nabla\cdot\frac{\nabla\phi_i}{U_\infty}} = \nabla\cdot(p\frac{\phi_i}{U_\infty}) \tag{8}$$

$$\frac{\nabla\phi_i}{U_\infty}\cdot\frac{\partial\vec{u}}{\partial t} = \nabla\cdot(\frac{\nabla\phi_i}{U_\infty}\frac{\partial\vec{u}}{\partial t}) - \cancel{\frac{\partial\vec{u}}{\partial t}\nabla\cdot\frac{\nabla\phi_i}{U_\infty}} = \nabla\cdot(\frac{\nabla\phi_i}{U_\infty}\frac{\partial\vec{u}}{\partial t}) \tag{9}$$

$$\frac{\nabla\phi_i}{U_\infty}\cdot\nabla(|\vec{u}|^2) = \nabla\cdot(|\vec{u}|^2\frac{\nabla\phi_i}{U_\infty}) - \cancel{|\vec{u}|^2\nabla\cdot\frac{\nabla\phi_i}{U_\infty}} = \nabla\cdot(|\vec{u}|^2\frac{\nabla\phi_i}{U_\infty}) \tag{10}$$

$$\frac{\nabla\phi_i}{U_\infty}\cdot(\nabla\times\vec{w}) = \nabla\cdot(\vec{w}\times\frac{\nabla\phi_i}{U_\infty}) - \cancel{\vec{w}\cdot\nabla\times\frac{\nabla\phi_i}{U_\infty}} = \nabla\cdot(\vec{w}\times\frac{\nabla\phi_i}{U_\infty}) \tag{11}$$

For this last Equation 11, applying again Equation 7, it is obtained a term which is calculated on the surface of the wings and will depend on the vorticity within the flow.

With all these steps, establishing a constant inflow velocity and knowing that $\vec{n}\cdot\nabla\phi = \vec{n}\cdot U_\infty\vec{e_i}$, two different expressions for the pressure have been achieved:

$$-\int_S p\frac{\nabla\phi_i}{U_\infty}\cdot\vec{n}dA = -\int_S \frac{\nabla\phi_i}{U_\infty}\frac{\partial\vec{u}}{\partial t}\cdot\vec{n}dA + \frac{1}{2}\int_S(|\vec{u}|^2)\vec{e_i}\cdot\vec{n}dA - \int_V \frac{\nabla\phi_i}{U_\infty}\cdot(\vec{u}\times\vec{w})dV + \nu\int_S \vec{n}\times\vec{w}\frac{\nabla\phi_i}{U_\infty}dA \tag{12}$$

$$\int_S p\frac{\nabla\phi_i}{U_\infty}\cdot\vec{n}dA = \int_S p\vec{n}\cdot\vec{e_i}dA \tag{13}$$

Furthermore, if the viscous shear contribution is added to the expression of the Total Force, the expression for the force derives in the following: $F_i = -\int_S p\vec{n}\cdot\vec{e_i}dA + \nu\int_S(\vec{w}\times\vec{n})\cdot\vec{e_i}dA$. Substituting the Equation 13 in this last identity, it is obtained the expression for the decomposition of forces which will be implemented during this project:

$$F_i = -\int_S \frac{\phi_i}{U_\infty}\frac{\partial\vec{u}}{\partial t}\cdot\vec{n}dA + \frac{1}{2}\int_S(|\vec{u}|)^2\vec{n}\cdot\vec{e_i}dA - \int_V(\vec{u}\times\vec{w})\cdot\frac{\nabla\phi_i}{U_\infty}dV + \nu\int_S(\vec{w}\times\vec{n})\cdot(\frac{\nabla\phi_i}{U_\infty}+\vec{e_i})dA \tag{14}$$

Now looking at each expression of Equation 14, the first two terms are considered as the body or added mass contribution (F_M) by which the inertial contributions are taken into account. The body term 1 considers the acceleration of the wing and the corresponding inertial contributions of the surrounding fluid as a consequence of the motion of the body. The body term 2 takes into account the surface velocity of the fluid.

The following term computes the vorticity within of the flow (F_V) in the whole domain of the control volume and the last term is the result of the surface vorticity (F_S). As it will be seen, this last part will be the unknown because the rest of them will be computed and F_i will be obtained from DNS.

For the computation of the added mass terms, it will be needed the velocity and acceleration of the wing. For this, it will be considered each node as a point particle so the kinematics equations of a particle will be applied to each of the surface wing points.

Considering the left wing in Figure 2b, an arbitrary point of the mesh with respect the non-inertial reference frame can be represented as $\vec{P_i} = p_x\vec{i}_w + p_y\vec{j}_w + p_z\vec{k}_w$. Translating this information to the non-moving dashed red line axis, the coordinates now are: $\vec{P_i} = p_x\vec{i}_w + (p_y - R)\vec{j}_w + p_z\vec{k}_w$ so the coordinate system will only have rotational motion equal to the change with time of Φ in the x direction. With this, the expresions for the velocity and acceleration:

$$\vec{u} = \frac{d\vec{P}\!\!\!/i}{\not{dt}} + \frac{d\vec{\Phi}}{dt}\times\vec{P}i \tag{15}$$

$$\frac{d\vec{u}}{dt} = \frac{d^2\vec{\Phi}}{dt^2}\times\vec{P}i + \frac{d\vec{\Phi}}{dt}\times(\frac{d\vec{\Phi}}{dt}\times\vec{P}i) \tag{16}$$

where $\frac{d\vec{\Phi}}{dt} = -\Phi_0 k sin(kt+\phi)\vec{i_w}$ and $\frac{d^2\vec{\Phi}}{dt^2} = -\Phi_0 k^2 cos(kt+\phi)\vec{i_w}$. Same procedure is followed for the right wing having into account the opposite sense of rotation.

Once this computations are performed for each of the surface points, the data is changed to the inertial reference. The final kinematics expressions are:

$$\vec{u} = |\frac{d\vec{\Phi}}{dt}|((py-R)\cdot cos(\Phi) - pz\cdot sin(\Phi))\vec{j} + ((py-R)\cdot sin(\Phi) + pz\cdot cos(\Phi))\vec{k}) \quad (17)$$

$$\begin{aligned}\frac{d\vec{u}}{dt} = (|\frac{d^2\vec{\Phi}}{dt^2}|((py-R)cos(\Phi) - (pz)sin(\Phi)) + |\frac{d\vec{\Phi}}{dt}|^2(-(py-R)sin(\Phi) - (pz)cos(\Phi)))\vec{j} + \\ +(|\frac{d^2\vec{\Phi}}{dt^2}|(py-R)sin(\Phi) + (pz)cos(\Phi)) + |\frac{d\vec{\Phi}}{dt}|^2((py-R)cos(\Phi) - (pz)sin(\Phi))\vec{k}\end{aligned} \quad (18)$$

where the angle $\Phi(t)$ is obtained from the Equation 1.

2.3 Code implementation

A Matlab code has been implemented for the computation of the forces which are associated with the flapping and heaving motion. Nevertheless, other software programs are used for helping in the computation of certain parameters in an efficiency way: *gmsh* and Fortran 77. The algorithm implemented with Matlab requires no user input besides the AR (for the geometry) and R (for the kinematics). Before starting with the deep study of the code, it will be presented in Figure 3 the flowchart of the overall process. It will be used as a reference for the explanations through the following sections.

The orange background will correspond with the Matlab computations, the blue one represents the *gmsh* part and the purple background are the processes performed by Fortran 77 in charge of carrying out BEMLIB libraries operations.

First, with the help of *gmsh* a mesh structure will be created with the model proposed. After that, using BEMLIB libraries the potential flow is calculated through Boundary Element Method. Finally with Matlab and taking information from the DNS performed with TUCAN, a control volume is created to calculate the interaction of the flow with the wings.

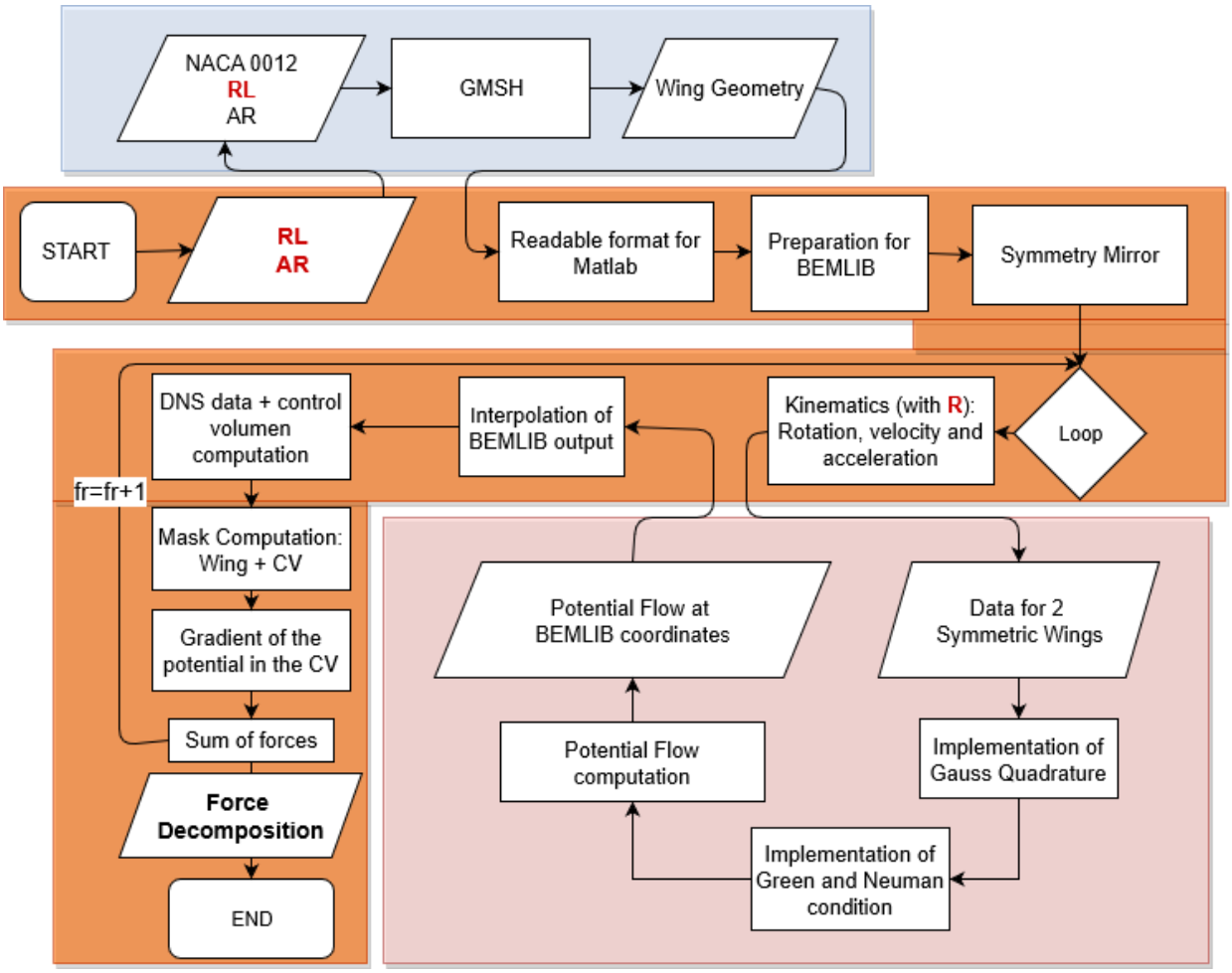


Figure 3: Total Flow Chart. User input represented in red

As expected, these processes are related to each other and all the inputs to the different software should be treated and prepared before being transformed into output from the corresponding program. Since the code is so dense and the time to do the project limited, it will be needed to optimize as much as possible the algorithms and tasks implemented.

Basically the code consists on the computation of the forces applied to the wing for different instants of time (frames). The scope of this project is reduced to study the behaviour of the aerodynamic parameters in one stroke (period), so that this stroke is divided in n frames to be studied separately through a Matlab loop. This period will be related with the reduced frequency k previously defined in Equation 2.

For all the cases studied during this project, the stroke has been divided into 32 frames (fr) so that each period has been divided into 32 equally instants of time. Furthermore, for better results, the data taken from DNS was saved from a certain initial time at which the forces computed started to converge. This convergence means that a stationary states was established.

Previous studies have demonstrated that the aerodynamic forces produced during a period of flapping motion are symmetric attending to the downstroke and upstroke [14]. So for the sake of simplicity and avoiding extra computations, it has been considered only downstroke process which corresponds with running 16 out of 32 frames.

2.4 Mesh

Parting from the Figure 3, the first part to be computed is the mesh using *gmsh*. The software *gmsh* is an open source meshing tool by which unstructured meshes are created. It consists on a 3D finite computational element with a programmed CAD motor and post-processor. Its main goal is to provide a quick and simple meshing tool with parametric input and advanced visualization skills. Through this software tool the model has been created from geometric inputs given in ASCII format. After this, the output from *gmsh* is converted from its own file format such that it is legible by Matlab. Thus a new mesh can be defined, created and given as input to BEMLIB.

These ASCII files works as a code for any NACA profile. It could be even adjusted for instance, the dihedral or twist angles by plotting the NACA profiles at different positions and angles. It is a helpful process since the code works with any geometry input. The desired model could be done using a *gmsh* graphical interface in which a wing is designed in CAD and automatically a code is generated and saved in a file. This file contains the Points, Lines and Curves from where the 3D wing will be meshed using *gmsh*. As mentioned, this output mesh data is introduced in to BEMLIB through Matlab .[28].

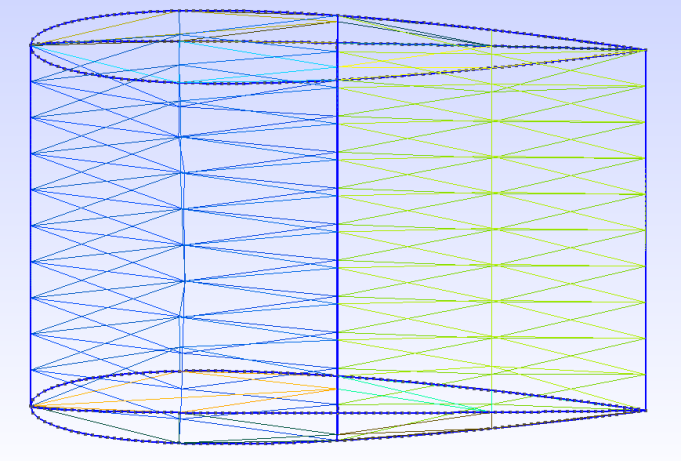


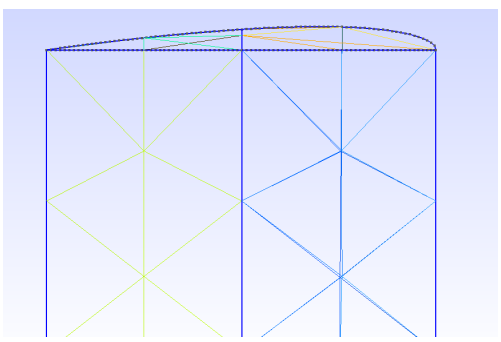
Figure 4: gmsh model from two NACA profiles

As it is observed in Figure 4, a triangular mesh is created from the plot of two NACA0012 profiles at the tips of the wing. From there, lines are created joining the key points: at leading and trailing edge, and depending on the mesh refinement, at $\frac{x}{2}$ for each of the corresponding subdivision. From there, nodes will create equilateral triangles through the main

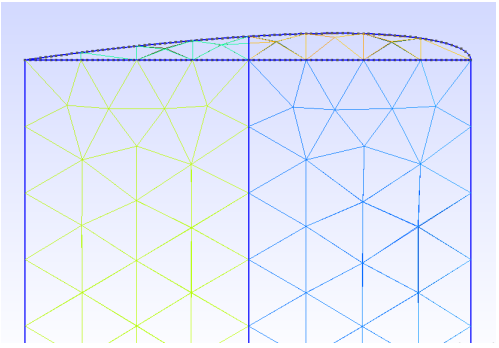
rectangles which has been previously designed.

Once algorithm has been specified, the software will compute a set of nodes which will create the required mesh shape. For this project, the Mesh algorithm selected is the Frontal algorithm [29]. It has been chosen this one because despite it is less robust and it requires more computational resources, a system of triangles close to equilateral shape are created simplifying the integration of the aerodynamic parameters calculated in the following steps. Since the mesh selected is not much bigger, the performance of the algorithm calculation is not much important. Furthermore, this algorithm will help on the interpolation of some parameters to the centroid of each equilateral triangle. It could be pointed that the Delauny was not selected because this triangulation often contains edges that do not satisfy the Delaunay condition. Curiously, a constrained Delaunay triangulation could not be a Delaunay triangulation itself and for some models this could be only checked by visual inspection [30]. So for simplicity, the Frontal method has been chosen.

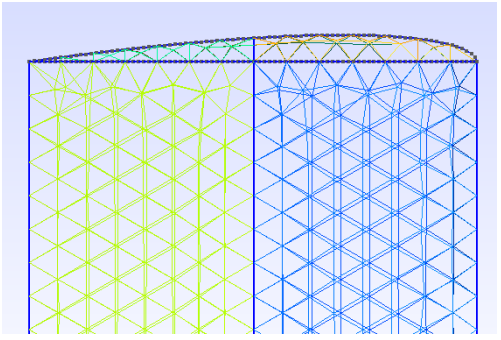
The refinement, so that the number of points created per wing, will be decisive: the more triangles there are, the longer it will take to calculate the aerodynamic parameters, but at the same time, the more reliable results are obtained. This is the reason why a trade off analysis for these constraints should be performed.



(a) RL=0.33 → 762 nodes



(b) RL=0.15 → 1594 nodes



(c) RL=0.075 → 3714 nodes

Figure 5: Number of Nodes depending on RL for an single wing of $AR = 2$

For a better understanding on how this works, different refinement has been presented in Figure 5 where it can be seen how, through the mentioned subdivisions, the nodes are

created. In each node it will be saved the necessary aerodynamic data related with the surface of the wing.

During this project, the parameter which relates the refinement of the mesh will be defined as Refinement Distance (RL). Using as support Figure 6, RL is defined as the distance between the centroid of each of the triangles located along the chord span. So for computing the number of points per chord for a given RL, it will be applied the following formula:

$$NP_{chord} = \left[\frac{c}{RL} + 2\right] \tag{19}$$

where c is the chord and NP_{chord} the number of points per chord.

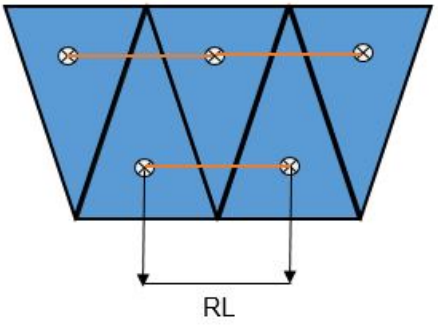


Figure 6: Refinement length for each triangle

Another important part of the wing modelling is the rounded shape at the tips as observed in Figure 7. It was created by adding the half of the NACA0012 profile in the Y=0 plane from each of the tips. After that, from the leading edge to the trailing edge different subdivisions are plotted from which the triangulation from the nodes will be done through this extra tip part. This design tries to model the shape of real wings avoiding as much as possible wing tip vortex.

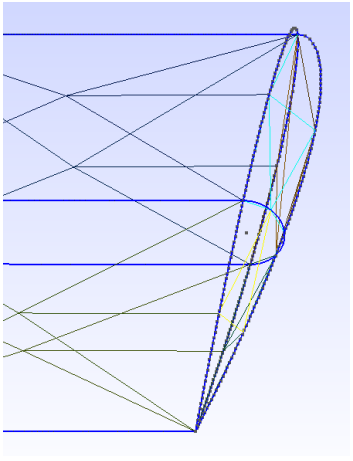


Figure 7: Rounded edges at the tips. (RL=0.5)

So during this project, the input data to *gmsh* software for creating the geometry of the wing are RL and AR. The final mesh created for one wing is plotted in Figure 8

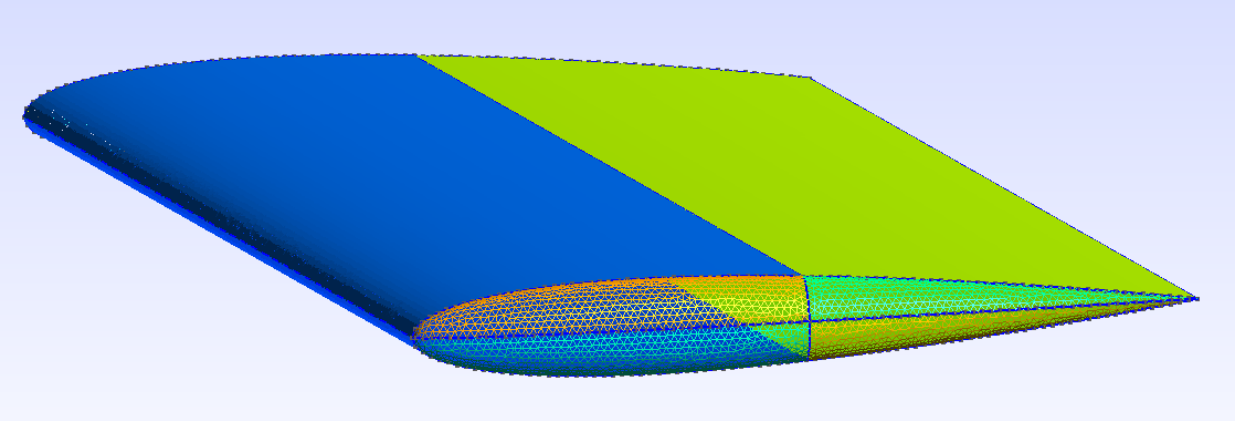


Figure 8: Mesh for $RL = 0.01$

As discussed, the *gmsh* software will create the mesh for only one wing. Then, the information computed by *gmsh* will be an input to Matlab where the geometry properties are converted into a readable Matlab array. Among these outputs, it can be highlighted the coordinates and normal vector for each of the nodes.

This part is not much decisive for the computation time of the whole code because as seen in Figure 3 it will be only performed once for each model and it does not last more than 30 seconds for a high refinement mesh.

2.5 BEMLIB pre-processing

Since the way the library handled the *gmsh* boundary conditions is unable to accept the ones required to calculate the potential, a pre-processing is required just before BEMLIB. For this reason, additional operations to the core code have been conducted.

Once the readable Matlab data is generated, the output data is edited for duplicating the points per triangle from 3 to 6 vertexes. This is because BEMLIB requires the triangular elements dened by 6 points instead of 3. For achieving this, a Matlab code has been implemented by which the triangular structures defined by three points are split resulting three extra points which are stored and added to each matrix data. The new geometric parameters are recalculated and updated. These new 3 points and 4 areas are created following the index order as it appears in Figure 9.

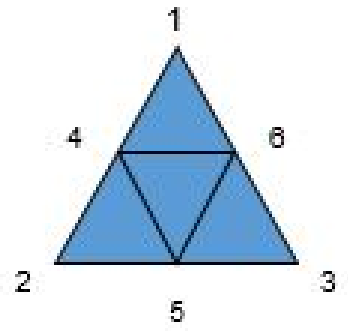


Figure 9: Pre-Processing for BEMLIB

In addition, there are cases in which the *gmsh* calculates the sense of the normal vectors in a random way. As previously introduced, the normal vectors are always pointing toward the fluid (the air). For this reason to apply the boundary conditions, a process in which all the vectors which does not satisfy this condition, are flipped.

First of all, for flipping these vectors, a point O_w inside the wing is computed. Taking advantage that the wing is completely convex, vectors formed between the mentioned point O_w and each point on the surface P, should be always pointing toward the fluid. So those vectors are created and saved and the scalar product between them and the normals is computed. For each position, if $\overrightarrow{O_wP} \cdot \vec{n} \geq 0$, it means the sense of the normal vector is correct. Otherwise the sense of this vector should be changed.

Until now, the code has only worked with the geometry of just one wing. Looking at the Flow Chart, after the information has been transformed in BEMLIB readable data, a mirror symmetric through the plane of symmetry is created. For performing this, a Matlab code will be implemented in such way that an identical wing is created making a mirror symmetry through the Y=0 plane. The information for the new wing is stored at the end of the geometric matrixes for the first wing.

So the parameters are ready for the computation of the potential flow inside the loop.

2.6 Potential flow computation: BEMLIB library

This Section probably is one of the most complex and longest part of the code so it will only be treated the most important aspects related with the potential flow. It will be talked about Neumann Boundary Conditions, Green's identity and it will be validated the numerical results comparing them with an analytical one. At the end of the Section, it will performed a grid refinement study.

The BEMLIB library consists in a set of Fortran 77 and Matlab files that solve the Laplace's equation through boundary element methods. By using this library, it could be

solved two and three dimensional geometries. One of the key point of using this discretization is that it is not required to solve the objective function in the whole domain, so that only in the required boundary. For the case to be analyzed, the potential flow over this geometry can be only solved where non penetration boundary condition is imposed.

The mentioned boundary element method is implemented for solving partial differential equations such as Laplace equation and Helmholtz's. For both wings, BEMLIB provides the mesh through spheroids which is the result of scaling a sphere in one or several axes for an exterior or interior potential. For the closed three-dimensional model proposed, the Neumannn boundary condition will specify the surface distribution of the normal derivative for the surface potential.

Furthermore, Paulete in [23] made certain changes to the original BEMLIB structure in order to adapt the tool to the decomposition requirements. First of all, the inputs to BEMLIB were eliminated by writing the conditions in a file with Matlab code and then been read by Fortran part before BEMLIB. Apart from it, originally the BEMLIB library was prepared for computing the potential in only one direction so that there was an influence matrix and one right hand side. The code was modified for obtaining one influence matrix and three right hand side. For this, the decomposition based on the algorithm for the linear system solver is able to obtain the solution in three directions even in less time than the required for the old code [31].

2.6.1 Neumann Boundary Conditions

Now it will be analyzed the Neumann Boundary Conditions imposed by BEMLIB for computing the potential flow.

As a consequence of the incompressibility character of the flow, from the continuity Equation 4 and the expression $\nabla\phi = \vec{u}$, the resultant identity gives $\nabla \cdot \nabla\phi = 0$ which corresponds with the Laplace equation.

The main objective for solving the Laplace equation proposed, is to find the value of potential through some volume of space. For this reason, it will be necessary to establish constraints on the unknown variable ϕ at the boundary surface of that space in order to obtain an unique solution [33]. For this there are two ways of implementing these boundary conditions: Neumann boundary conditions and Dirichlet Boundary Conditions.

Through Neumann boundary condition in the Laplace equation it is imposed that the directional derivative of ϕ is some value at some location. This condition affirms that the directional derivative normal to some boundary surface, called the normal derivative, is equal to the projection of the free stream velocity in the normal vector to the surface. Summarizing the equations implemented for the surface potential computation it is obtained that:

$$\begin{aligned} \nabla^2\phi_i = 0, \text{ In all V} \\ \nabla\phi_i \cdot \vec{n} = -\vec{n} \cdot \vec{e_i}\, U_\infty, \text{ At the wing surface} \\ \phi_i \to 0, \text{At i} \to \infty \end{aligned} \tag{20}$$

Where the first line corresponds with Laplace equation, the second one is the Neuman boundary condition and the last equation ensures that there are no perturbations on the flow at infinity. The vector $\vec{n}$ represents the vector normal to the surface of the wings.

2.6.2 Green's identity

Continuing with the analysis with the Fortran code, for solving the Neumann boundary condition numerically, the Green's third identity method has been implemented at the boundary through the following equation: [27]

$$f(\vec{x_0}) = \int_S [-\nabla f(\vec{x})\; G(\vec{x}, \vec{x_0}) + f(\vec{x})\; \nabla G(\vec{x}, \vec{x_0})\;] \cdot \vec{n}(\vec{x})\; dA(\vec{x}), \tag{21}$$

Where the point x_0 lies in the solution domain (referred to wing surface S) and $G(\vec{x}, \vec{x_0}) = \frac{1}{4\pi|\vec{x}-\vec{x_0}|}$ is the free-space Green's function of Laplace equation in three dimensions.

So taking the limit of the domain, as x_0 approaches the surface S and declaring the limit of the double-layer potential as a function of its main value, it has been obtained an integral identity of the second kind for the surface distribution of $f(\vec{x})$.

$$\int_S^{PV} f(\vec{x})\; [\; \vec{n}(\vec{x}) \cdot \nabla G(\vec{x}, \vec{x_0})\;]\; dA(\vec{x}) - \frac{1}{2} f(\vec{x_0}) = \pm \int_S G(\vec{x}, \vec{x_0})\; [\; \vec{n}(\vec{x}) \cdot \nabla f(\vec{x})\;]\; dA(\vec{x}) \tag{22}$$

where PV is set to be the principal value of the double-layer integral. It should be pointed that when the Equation 22 is applied for the interior problem, there exist infinite number of solutions that differ by an arbitrary constant. Otherwise, since the required potential for this project will be computed at the surface of the wings (so that exterior), the solution will be unique.

2.6.3 Kinematics and Geometric dependency of the potential flow

It was decided to compute for each Cartesian coordinate the surface potential. The idea of that was to calculate the surface potential only once per stroke, and through the projections in respective axis calculate the surface potential for each rotation using:

$$\phi_\alpha = \alpha_1\phi_x + \alpha_2\phi_y + \alpha_3\phi_z \tag{23}$$

where the parameter $\vec{\alpha} = \alpha_1\vec{e_x} + \alpha_2\vec{e_y} + \alpha_3\vec{e_z}$ represents the angle of attack of the free

stream to the wings in the three Cartesian coordinates. This last expression demonstrates the linearity of the potential with the orientation of the body [6]. This way of representing the potential is useful since it allows to calculate the potential only once and then projecting the corresponding inflow with the given angle α for each frame.

Here it comes the first important decision about the code implementation. This method saves a lot of computational time because the surface potential is only computed once and the angle of attack α is changed for each frame so that there is a different ϕ_α for each instant of time. For the case of study, this method is not possible because the relative position of both wings changes with the angle Φ ($\pm\Phi$, respectively) so that the Equation 23 cannot be applied.

Going deeply, in Figure 2b, it is represented how is going to change the angle of attack of the wing through this process. As it could be observed, for applying the Equation 23 it is necessary that the components of Φ affects equally to both wings, so that their relative position does not change with time. This fact could be reached for instance if Φ were the bank angle for a conventional airplane turn. [34]

So the solution proposed for this issue is to calculate the geometry, create the respective rotations of the wing along the x_w axis, and then compute the potential flow inside the loop.

The final question for the computation of the potential flow was to decide if input one wing and duplicate the potential flow to the second wing or if input both wings to BEMLIB libraries obtaining the flow field for both. As mentioned, the potential flow will depend on the geometry and the angle of attack of the wing. The advantage of inputting only one is to reduce by half the computation time for this part. This is why multiple tests have been performed for seeing the influence of the second wing on the first one in terms of the potential flow. The conclusion figured out is that due to the close distance at which they are mounted, the influence that one wing have on the other one is enough so that both wings should be introduced in BEMLIB part.

After the solution is obtained from BEMLIB, two separated interpolations are performed with the information of each wing. This is because the coordinates for the BEMLIB output parameters do not match with the input data from Matlab: as introduced BEMLIB mesh works with spheroids and the *gmsh* mesh output were equilateral triangles. As convenience, equilateral triangles will be used during the rest of the code.

2.6.4 Validation of BEMLIB libraries

Before continuing with the code analysis, the BEMLIB libraries should be checked so that Fortran part computes correctly the potential flow. This means that the libraries are correct set following the indications posted in Pozrikidis guide. [27]

For this, it will be computed the analytical solution of an sphere and it will be compared with the solution obtained through BEMLIB. The analytical solution for potential flow only exists for simple shapes as circles or ellipses in 2D. As the BEMLIB is configured for 3D cases, it has been chosen the simplest 3D shape which is the sphere.

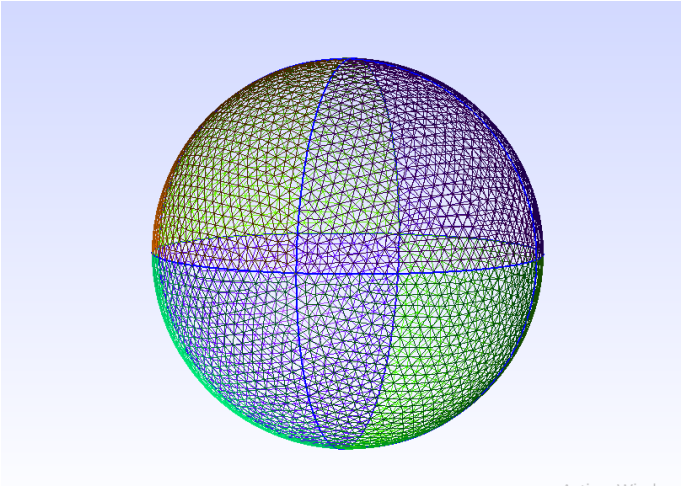


Figure 10: Sphere mesh created for checking BEMLIB libraries

As could be observed in Figure 10, through *gmsh*, a spherical mesh of unitary radius has been obtained. After that, the code will run and calculate the corresponding potential flow using BEMLIB.

On the other hand, the analytical expression is calculated from Equation 20. The Laplacian presented will be solved through separation of variables, choosing for convenience a reference frame attached to the center of the sphere [11]. Since the sphere presents symmetric flow with respect Y and Z axis, the expressions for these axis will be the same.

For this part, it will be only presented the potential flow through x axis since U_∞ is always pointing in the positive x-axis direction during this project.

$$\phi_x(x,y,z) = U_\infty x\Big(1 + \frac{R^3}{2(x^2+y^2+z^2)^{\frac{3}{2}}}\Big) \tag{24}$$

Analyzing the analytical expression, the velocity field through the grid is obtained from the formula $\vec{U} = \vec{U_\infty} + \vec{u'}$ where $\vec{u'}$ is the perturbation velocity. BEMLIB libraries only computes the potential flow related with the perturbation part, that looking at Equation 24, it would correspond with the right hand side term. So for obtaining the same field using both methods (numerical/analytical), it should be added the expression $U_\infty x$ (from the left hand side of Equation 24) to the BEMLIB results.

With this, both solutions has been plotted and it is observed that the results are exactly the same with smooth discrepancies at the begining and at the end of the x-axis.

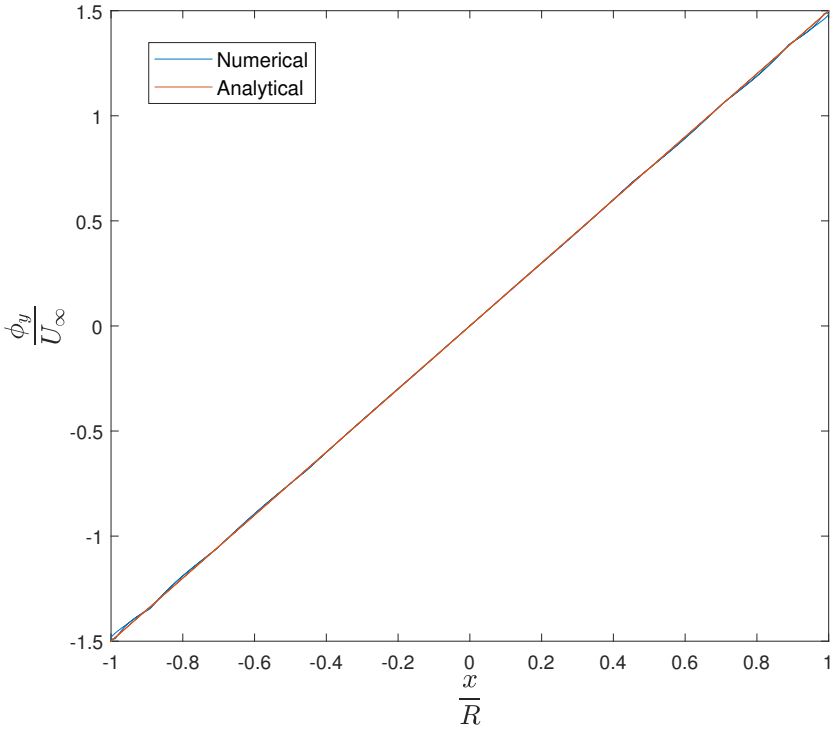


Figure 11: Potential flow analysis. $\frac{\phi_x}{U_\infty}$ vs $\frac{x}{R}$ using numerical and analytical methods

Same results are obtained for z and y directions so it can be concluded that the BEMLIB libraries are correct set.

2.6.5 Mesh Refinement Analysis

Finally, a mesh study is performed during this Section to determine the optimum number of points in a time-error trade off. For doing this, the potential flow will be computed for different refinement meshes (so different RL). These different RL are summarised in the following table with the corresponding points created for AR=2:

RL	0.01	0.05	0.07	0.075	0.08	0.085	0.09	0.095	0.1	0.15	0.20	0.25	0.3
Points (per wing)	191418	8182	4366	3926	3598	3250	2798	2498	2370	1594	1266	986	762

Table 1: Number of points meshed per wing, for each RL. AR=2

The constraints imposed for achieving this is to find the optimum refinement mesh which will give the best accuracy results. So it was decided to calculate the surface potential in several stages of mesh refinement and compare the results with the most refined mesh which is assumed to be the most accurate.

The most accurate one is considered to be $RL = 0.01$ whose mesh appears in Figure 8. This case is not possible to be developed for the project due to the prohibitive time it would take for solving the BEMLIB part for each frame.

Due to the variation of the number of points for each of the stages, the results for each RL are obtained at different coordinates with respect the most refinement one. For this reason an interpolation to the points of $RL = 0.01$ has been performed and the average error has been computed. Parallel, the time needed for the computation has been obtained. These three parameters are plotted in the following Figure:

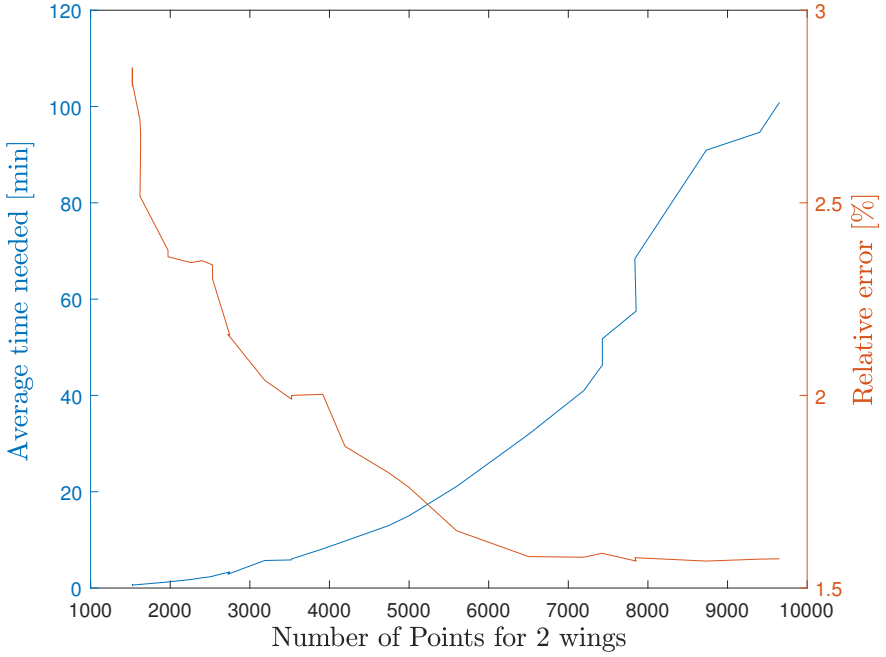


Figure 12: Mesh Refinement analysis for 2 wings with AR=2

As can be observed, there is a part in the graph in which the relative error is maintained constant with the mesh refinement, but the needed time grows with the number of points. Taking into advantage this fact, it has been chosen the part of the graph at which the error is constant with the refinement and the computational time is the minimum. This was observed to be 7852 points on the surface of both wings with an average error of 1.65% for computing the potential flow.

So this refinement will correspond with a RL=0.075 and 3926 points per wing with AR=2. Coming back to the Equation 19, the number of points per chord will be $NP_{chord} = [15.33] = 15$. Maintaining the same RL when AR=4, the resultant mesh will be 7234 points per wing.

So once the mesh is selected, the potential flow ϕ is computed for each of the points on

the surface of the wing. This parameter will be used for computing the body mass term 1 and the potential gradient for the control volume term in the decomposition force from Equation 14.

2.7 DNS analysis

After the surface potential has been computed, the wings will be introduced into the corresponding control volume with the aerodynamic information needed for computing the forces. The required flow data is read from a DNS performed in the CFD group of the Bioengineering and Aerospace Engineering department of Universidad Carlos III de Madrid. This DNS data was computed through TUCAN software.

The method used in TUCAN is the immersed boundary method applied to the full Navier-Stokes equations and assuming an incompressible flow [6]. Through this process two isolated meshes are involved: the first mesh defines an object in the flow (the wing) and the second one defines the control volume. The presence of the wing on this system is designed through direct forcing immersed boundary method proposed by Uhlmann [26]. As previously mentioned, the refinement used for this control volume data is 56 points per chord. This mesh refinement is still far from the one obtained in last section for the wing surface (15 points).

Since both meshes are constructed separately, the points on both of them, do not necessarily match with each other and there are volume points inside the mentioned object. In principle, there are no coordinates on the control volume mesh on which the boundary conditions can be directly imposed. For this reason, some constraints are set at the surface. This leads to solutions where the flow variables are smeared at the body boundary.

Continuing with the flow chart, this section is in charge of computing the volume integral term in the Total decomposition Equation 14. It corresponds with F_V and is defined as:

$$F_V = -\int_V (\vec{u} \times \vec{w}) \cdot \frac{\nabla \phi}{U_\infty} dV \tag{25}$$

so the parameters needed for computing it will be the velocity field $\vec{u}$, the vorticity field $\vec{w}$ and the gradient of the potential for the control volume $\frac{\nabla \phi}{U_\infty}$. During this Section, it will be explained the derivation of these parameters.

2.7.1 Control Volume data

The control volume in which the wing will be located, is divided in small cubes and the DNS data will be located at different coordinates for each cube. If a cut is done to the Control Volume, the face of each cube can be seen in Figure 13, whose configuration is

known as staggered grid. So the data obtained from TUCAN and used in Matlab is the following:

- **Pressure Points:** These coordinates represent the center of the cube and no information about pressure or any other parameter will be output.
- **Velocity Field:** These values are located at the middle of the edge of each small cube. As it appears in Figure 13, the red and blue triangles represents the x and y velocity direction respectively.
- **Vorticity Field:** These parameters come from the velocity field using second order spatial finite differences. The results of this computation are situated at the corners of each cube.

The Total Force applied to the wing is also obtained from DNS and it will neccesary for the computation of the F_S term presented in Equation 14.

The flow data corresponds with one wing so during the rest of the code, it will be treated the forces applied to the right wing. Nevertheless, the second wing will have certain influence on some of the decomposed forces.

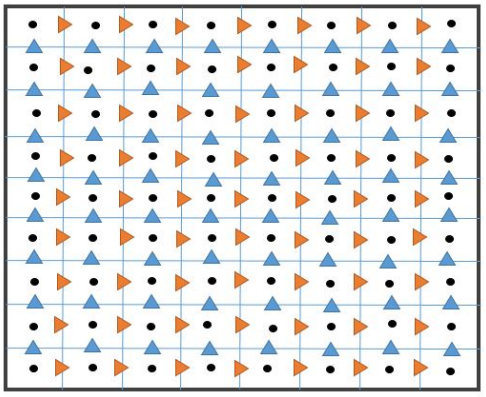


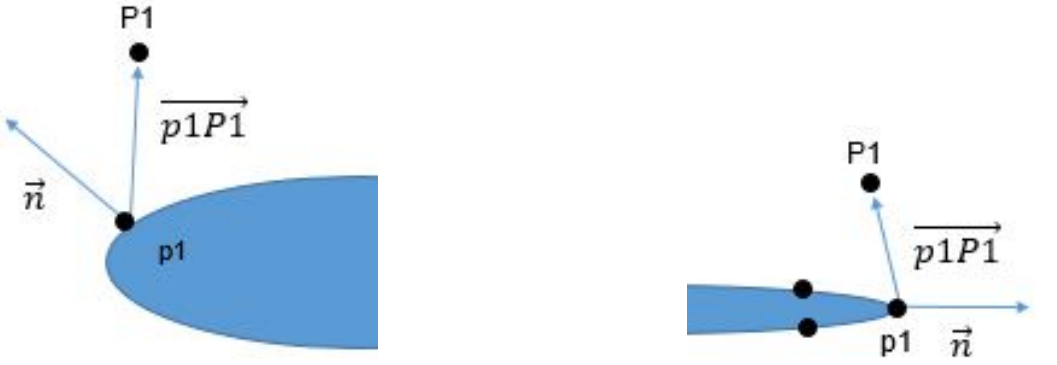
Figure 13: Snapshot of the Control volume field: Staggered Grid

It is necessary to have the information for each aerodynamic parameter at common points because this will help to solve properly the equation of forces for each contribution. For this reason, the data previously introduced will be linearly interpolated at the Pressure Points. For doing it, the 4 vorticity values of each cell are averaged at the center of the cell. On the other hand for the velocity, the values of opposing faces are averaged at the center of the face cube.

2.7.2 Mask computation

After DNS data is obtained, a routine has been created to separate the control volume points from the wing surface ones. The output for this task is a multi-array with 1 for the values of the control voulume which are outside the wing, and with 0 for those values of the control volume which are inside.

For doing it, a routine is implemented between the points of the surface of the wing (with their normals as well) and the pressure coordinates. This routine works in the following way:



(a) Leading Edge Computation (b) Trailing Edge Computation

Figure 14: inout Algorithm

First of all, a bounding box around the wing has been defined in order to avoid unnecessary computations so that the algorithm is only applied to those points inside the bounding box. This small step saves around 80% to 90% of the time that this process could take if is not applied. After that, an array is created with the nearest points of the control volume to the wing points.

Looking at Figure 14a, a random point P1 from the control volume has been taken. For knowing if it is inside or outside the wing, it will be computed a vector which goes from this point to P1: $\overrightarrow{p1P1}$. If the dot product between this vector and the normal in p1 is positive, it means that P1 is outside. Nevertheless if the result is negative, it means that the point P1 is inside.

It should be mentioned that this algorithm works when the curvature on the surface is not much large compared with the density of surface points. If this happens, it can deal to the fact that the angle between $\overrightarrow{p1P1}$ and $\vec{n}$ is larger than 90º. So it will act as if P1 is inside the wing. Looking at Figure 14b, this last circumstance can happen at the trailing edge where a low mesh refinement can create angles bigger than 90º.

The solution obtained has been plotted in Fgure 15, where the cyan points represents 1 and white corresponds with 0. The NACA 0012 profile with 15 points per chord has been also plotted to see the mask created for both meshes. It can be seen the importance of the

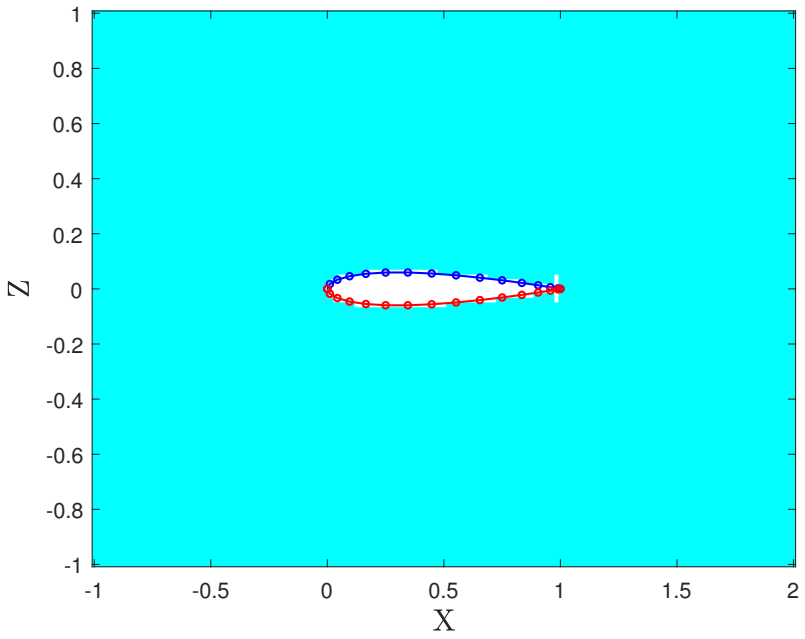


Figure 15: Midspan mask for the wing

refinement for the wing mesh since as explained, it can appear errors at those zones where the curvature is larger than points density: at the trailing edge.

2.7.3 Potential Gradient: computation methods and optimization

For computing the volume integral proposed for Equation 25, it will be needed the gradient of the potential. Now it will be presented two different ways of computing the gradient and they will be compared with each other presenting their corresponding advantages and disadvantages.

- **Gradient by finite differences:** Through this method, the potential in the control volume is computed at each cell center by solving the Green's third identity used in BEMLIB with Equation 21. For this case, the parameter $\vec{x}$ represents the position vector of the point on the surface of the wing and $\vec{x_0}$ represents the position vector of the point in the control volume cell. So the value $\vec{r}$ which appears in the term $G(\vec{x}, \vec{x_0})$ represents the distance between those two points. After the potential value has been calculated on the wing points, the gradient for the control volume is computed by second-order spatial differences.

- **Gradient directly:** It could be also calculated by using a formula which directly gives the gradient of the potential:

$$\frac{\delta f(\vec{x_0})}{\delta x_{0i}} = \frac{1}{4\pi}\int_S \frac{x_i}{r^3}[\vec{n}(\vec{x})\nabla f(\vec{x})]dA(\vec{x}) + \sum_{j=1}^{3}\frac{1}{4\pi}\int_S f(\vec{x})\Big(\frac{\delta_{ij}}{r^3} - 3\frac{x_i x_j}{r^5}\Big)\vec{n_j}(\vec{x})dA(\vec{x}) \quad (26)$$

where the subindexes i and j denote the x y z directions and δ_{ij} is Kronecker's delta. This component is a mathematical equivalent which works in the following way: [35]

$$\delta_{ij} = \begin{cases} \text{if i}\neq j \\ 1 \text{ if i=j} \end{cases} \quad (27)$$

It is known than using this last Equation, the results will be more accurate but due to the complexity this process presents, it will need a high amount of time.

On the other hand, through first method, it will output results with errors associated to the finite difference. Furthermore, the data is computed at the half of the distance between each cell center so a linear interpolation should be performed adding another error to the process.

Despite the second process takes twice as long as the first one, it was decided to implement it. Nevertheless, it will be reduced as much as possible the computation time by modifying some aspects in the code.

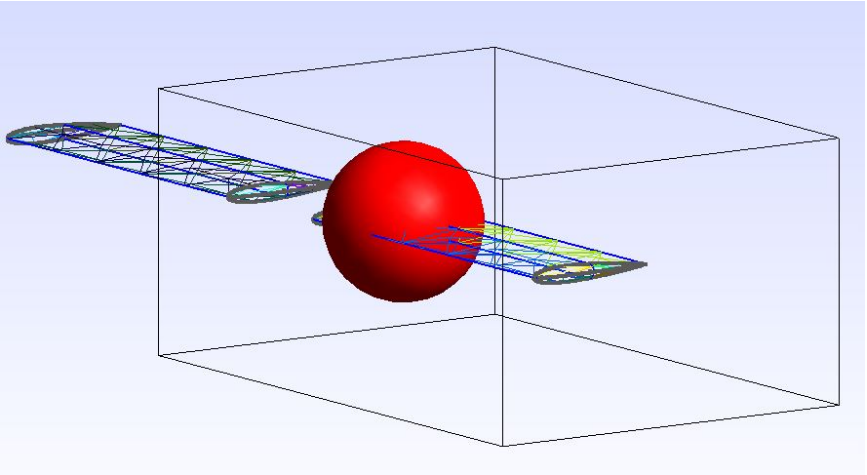


Figure 16: Potential Gradient Optimization by creating an sphere

For explaining this task, a brief analysis of the implementation of Equation 26 should be done. This Equation calculates the influence that each of the wing subtriangles that appear in Figure 9 have on each control volume point. The six points are taken into account for computing the potential gradient on the control volume. Looking at Equation 26, the influence of each point decays with the distance a rate of fifth power so there will be a distance at which the influence will be almost negligible (concretely on the order of

$1e^{-10}$). This value will have a low influence on the gradient results. For this reason a kind of extra control volume has been created with the shape of a sphere and center situated on the centroid of each node (Figure 16). The potential gradient of the CV points which are inside this control volume will be calculated using the six wing points previously mentioned. On the other hand, the potential gradient of the CV points which are outside the sphere, will be computed by using the centroid of each wing triangle so that all the six parameters need to be interpolated to this point.

The optimum Radius of this sphere (R_{opt}) will be obtained by doing a routine for calculating the minimum radius for which the results obtained for F_V slightly vary with respect a model with a sphere of infinite R_{opt}.

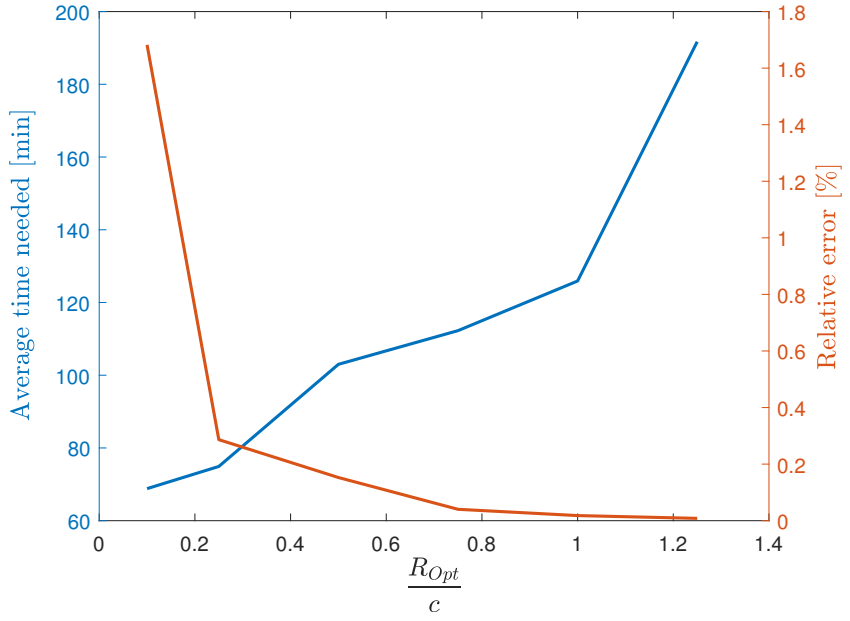


Figure 17: Potential Gradient Optimization by creating an sphere

As can be observed, the results are similar to the ones obtained in Section 2.6.5 so the same procedure is followed: it will be used the R_{opt} with the best trade-off time-error. Looking at the Figure this will correspond with $R_{opt} = c$, for which it will be a 0.05% of error. With this optimization, more than 50% of computational time has been saved.

2.7.4 Control Volume Size

The size of the CV has been set to be 1 chord from the maximum and minimum values from each of the three coordinates of the wing points. Here it appears the second decisive point for the code implementation. It should be chosen if the CV size will be computed taken into account 1 or 2 wings. If AR=2 and $\Phi = 0$, for the first option the dimensions would be 3cx3.5cx2c (negative values of y-axis have not been included), and for the second

one 3cx7cx2c (the y-direction doubles).

Duplicating the size could derive in problems with the memory of the system and with a prohibitive increase of computation time. Finally, the decision carried out was to implement the CV for the right wing, but taking into account the left wing information as well. This corresponds with the left wing points for computing the potential gradient.

In conclusion, the total size is 3cx3.5cx2c for $AR = 2$ and 3cx5.5cx2c for $AR = 4$.

Once the kinematics parameters, the potential on the surface of the wing and the potential gradient in the control valume has been obtained, it will be computed each of the contribution of the forces. As the total force is obtained from DNS, the surface vorticity in Equation 14 is calculated from:

$$\vec{F}_S = \vec{F}_T - (\vec{F}_M + \vec{F}_V) \tag{28}$$

2.8 Models to be studied

So for finishing the Methodology Section, here are the models to be analyzed during Results Section:

Nomenclature	Wings	AR	Radius of Rotation (R)	Φ_0 (rad)
AR2-1H	1	2	∞	0
AR2-2H	2	2	∞	0
AR2-R8	2	2	8	0.10017
AR2-R2	2	2	2	0.25268
AR2-R0	2	2	0	$\pi/6$
AR4-R0	2	4	0	0.25268

Table 2: Models to be studied

All models, except AR=4, will have the same geometry. The difference between them will be marked by the R which will change the kinematics of the motion. For the first two models, the difference will be in the number of wings.

3 Results

Through this Section, it will be used the modified code for comparing the unsteady aerodynamics which it appears during the simulation of the models presented in Table 2.

First part will treat the computational time needed for each model. After that it will be studied the variation in the results when 1 or 2 wings are simulated. Then, it will be studied the influence on the unsteady aerodynamics that the kinematics and the geometric properties have in a forward flapping motion. For this, it will be compared the consequences on the forces when R is changed and through last pages, a value of R has been fixed to 0 and AR has been modified from 2 to 4.

As explained, the decomposed forces will be obtained for the downstroke which corresponds with the half of a Period T (from $\frac{t}{T} = 0$ to $\frac{t}{T} = 0.5$). The same analysis and conclusions would be obtained for the upstroke motion which goes from $\frac{t}{T} = 0.5$ to $\frac{t}{T} = 1$. The middle of the downstroke $\frac{t}{T} = 0.25$ will be taken as reference for the computation and comparison of certain variables between the models, since it corresponds with the maximum vertical velocity of the body.

The data to be input is $h_0 = 1$ and $c = 1$ so the flapping amplitude for all the cases is 1 (as explained this will constraint the wing tip vertical velocity). The variables Φ_m and φ are 0. During all the cases studied, it has been considered a reduced frequency $k = 1$, so $T = 1$. For the Reynold number $Re = \frac{U_\infty c}{\nu} = 500$, which it would correspond with a relative small device. The free stream velocity will be $U_\infty = 1m/s$.

For comparing the results, it should be defined the thrust and lift densities which are representatives variables for the distribution of these aerodynamic parameters. They are defined as:

$$\delta_t = \frac{2(\vec{u} \times \vec{w}) \cdot \nabla \phi_x}{U_\infty^3 cb} \tag{29}$$

$$\delta_l = -\frac{2(\vec{u} \times \vec{w}) \cdot \nabla \phi_z}{U_\infty^3 cb} \tag{30}$$

3.1 Computational Time required

In the next Table, there will be presented the error and average time needed for the simulation of two flapping wings with $AR = 2$. The corresponding control will be 3cx3.5cx2c. This wing configuration, will have a mesh wing refinement of 7852 points (both wings included) and a $R_{opt} = c$ for the control volume computations. For obtaining the errors, the output results are compared with the most refinement case for the mesh operation and with a R_{opt} enough bigger for the CV optimization. For the errors in the mask computation at the trailing edge in Figure 15, it has been considered that these points represent about 1%

of the total control volume. So this proportion has been considered as the error for this code part.

	Mesh (*gmsh*)	Pontential Flow (BEMLIB)	in/out mask (Matlab)	Potential Gradient (Matlab)
Time (min)	0.5	60	5	200
Error (%)	0	1.7	1	0.05

Table 3: Time and Errors associated to the Code

Looking at Table 3, there is a 0 error for the Frontal algorithm implemented by *gmsh* software, assuming a perfect equilateral triangulation for the chosen mesh refinement.

The time results are considered by frame so the required time required for computing the downstroke forces per model (16 frames) will be 265.5x16=4248min (70 hours). This makes a total time of 3 days per model. For the case of AR=4, as the control volume size increases to 3cx5.5cx2c and 14468 wing surface points , the model computational is 5 days.

3.2 Heaving Case: 1 Wing vs 2 Wings

During this Section it will be studied the consequences that the presence of a second wing can have on the decomposition forces. It will be analyzed the heaving motion of one and two wings for the same simulation conditions. The DNS data taken corresponds with the simulation of two wings in heaving motion. Looking at Table 2, both heaving motions are achieved by imposing an infinite R which leads to $\Phi_0 = 0$.

Before starting with the analysis of the force decomposition, it has been plotted the mapping of vorticity for $\frac{t}{T} = 0.25$ and 50% of the midspan.

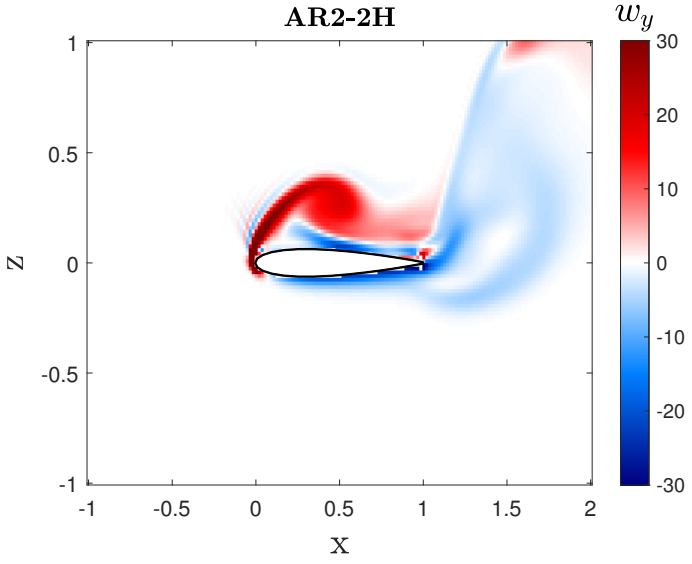


Figure 18: w_y field at the middle of downstroke and mid-span

As observed in Figure 18, during a heaving fight, the principal vortex is created at the leading edge (LE), where the vortex trails behind the wing as it moves and separates from LE. It is known that at the middle of the downstroke, the leading edge vortex (LEV) is the greatest since the wing will have the highest down speed. The LEV will be the main contributor for this analysis since the fast change in local angle of attack will produce a low pressure region inside the LEV so it will be the main lift enhancing mechanism of flapping wings. If the wing were fixed, this process of lift generation is associated with the dynamic stall in which a sudden drop of the aerodynamic force is produced. But the cyclic motion generates a transient lift due to a LEV which avoids entering in the dynamic stall previously mentioned [5]. Furthermore, it has been demonstrated that the larger thrust efficiency is achieved for a perfect interaction between the LEV and the trailing edge vortex (TEV) producing two vortices per stroke [6].

Continuing with the heaving comparison, each of the term from decomposition Equation 14 have been summarized in Figures 19 and 20. In a quick view, it could be observed that both plots follow the same path with small differences at the first and last stages of the downstroke.

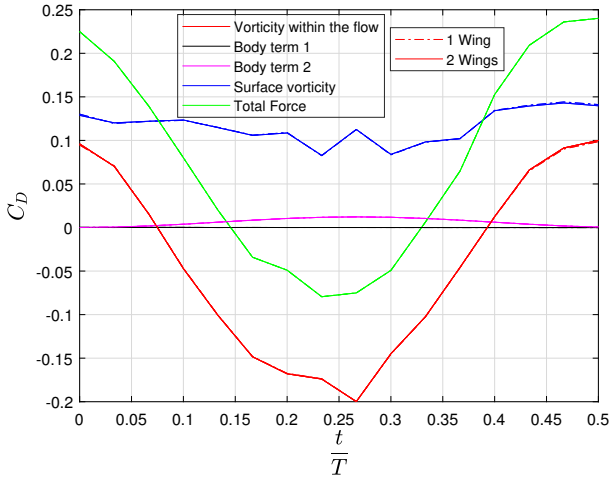


Figure 19: Drag Coefficient C_D

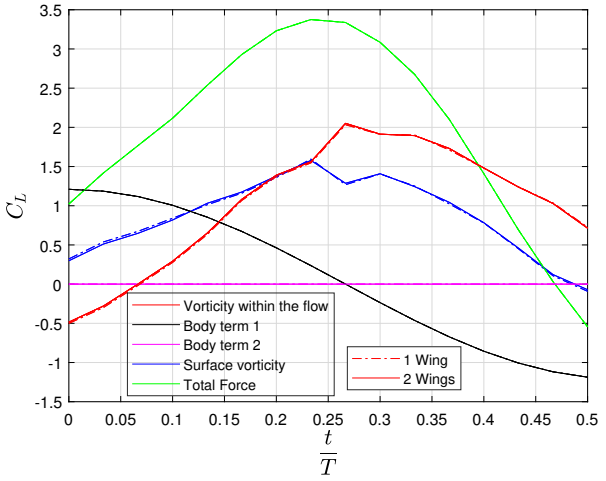


Figure 20: Lift Coefficient C_L

The Total Force is also plotted to see the fluctuation that the wing suffer during half of stroke. Looking at Figure 20, when the wing goes down, lift appears, so intuitively can be guessed that for the upstroke negative lift will appear. On the other hand, for Figure 19, the drag for the upstroke would follow the same path as in the downstroke. The total force, obtained from the DNS study, is the sum of the other four lines: the surface vorticity (F_S), the vorticity within the flow (F_V) and the added mass terms (F_M).

The F_M term, as is known, is the consequence of the fluid motion due to the change in position of the wing. At the beginning of the downstroke, the wing will have the maximum acceleration and the minimum velocity so the fluid surrounding the body will remain at

rest resisting the acceleration. When $\frac{t}{T} = 0.25$, the air moves down with the body and deceleration appears with the consequence that the fluid hits the wing producing a negative force. Looking at Figures 19 and 20, for C_L plot there is a higher influence from the body term than for C_D plot because the wing will move more air mass through the z-axis than through the x axis due to the bigger surface projection on this axis. With this it can be affirmed that there is scarcely dragged fluid in the x-direction.

The next term to discuss is F_V. As seen in Equation 25, this term depends on the potential gradient of the flow in the control volume. The computation of this parameter comes from Equation 26 where it can be observed that it decreases with the inverse of the distance to the wing to the fifth power. So close vortex structures to the wing will be decisive for this contribution. On both plots, it could be observed that the maximum value for this term is achieved approximately at maximum velocity when $\frac{t}{T} = 0.25$ with a certain lag due to the time the vortex needs to be created.

The last contribution to be analyzed is F_S which is the result of the forces acting on the body tangentially to its surface. Looking at C_D plot, it can be observed that this term is the principal contributor for the creation of drag. This drag is balanced by F_V which creates from $\frac{t}{T} = 0.15$ to $\frac{t}{T} \approx 0.34$ a negative value which means thrust.

Now it will be analyzed the differences between 1H and 2H. As previously mentioned, there are some discrepancies for the potential flow ϕ of right wing when it is presented alone and when there is a second wing on the left hand side. Logically, this discrepancies will also create differences for the computation of the potential gradient $\nabla\phi$.

Attending to the body term 1, it could be seen that it depends on ϕ. For the C_L plot in Figure 30, it could be observed that for both models, the results are almost the same with a maximum difference of 0.89%. Since the body term 2 only depends on the kinematics of the right wing, the results for both cases are exactly the same. With this, it could be affirmed that the differences in the potential flow between both models, won't decisively affect the results.

For F_V term, maximum relative error has been set to be 1% in the plot C_L and 4% for the F_S in the plot C_D. With this, it can be also declared that despite the gradient of the potential flow is different for both models, the results will be very similar.

The reason for this high similarity is that for the DNS flow presented, there is no instabilities through the spanwise axis. With this, it could be affirmed that for simulating a pair of symmetric wing system, it could be presented just the aerodynamic data for one wing and the DNS information for two wings. Nevertheless, if there were inestabilities through the spanwise direction, the second wing data should be taken into account.

Finally, from Equations 29 and 30, it can be plotted a map for densities terms in the middle of the downstroke and situated at the mid-span of the wing. From these plots, it can be observed how the density barely changes from one model to other.

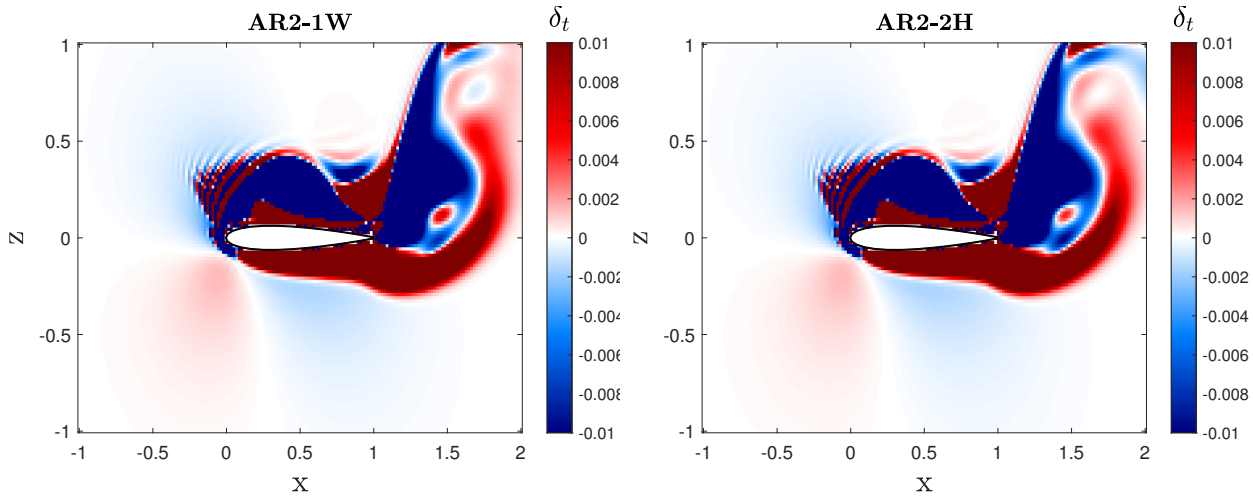


Figure 21: δ_t for Heaving Motion for 1 and 2 Wings

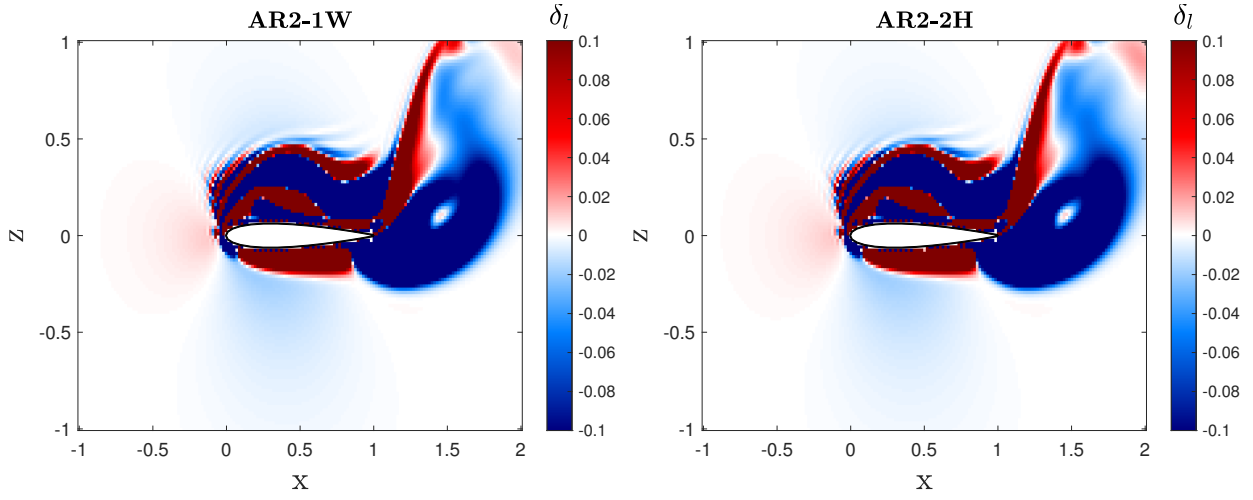


Figure 22: δ_l for Heaving Motion for 1 and 2 Wings

As it could be observed, the results are almost the same. For a better understanding on how this variable is obtained, it has been plotted in Figures 29 and 30 the velocity fields in the vertical and x-direction. These fields, multiplied by the map vortex in Figure 18, give as output the results for the densities in Figure 21 and 22. The smooth differences that it appears are because the potential gradient part, whose influence has demonstrated to be negligible for the forces results.

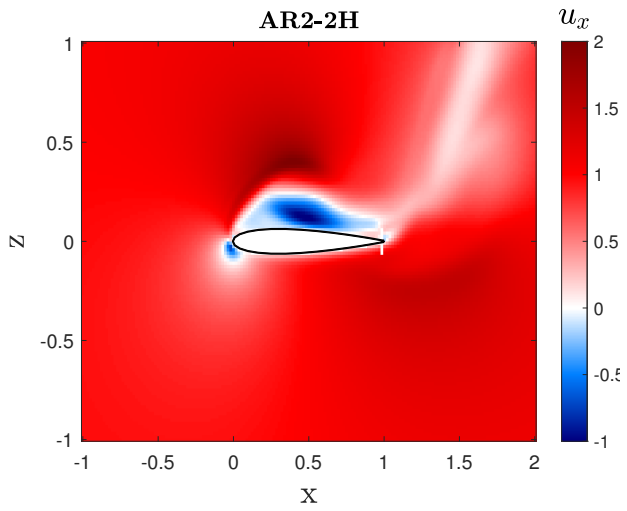


Figure 23: u_x field (50% spwise) ($\frac{t}{T} = 0.25$)

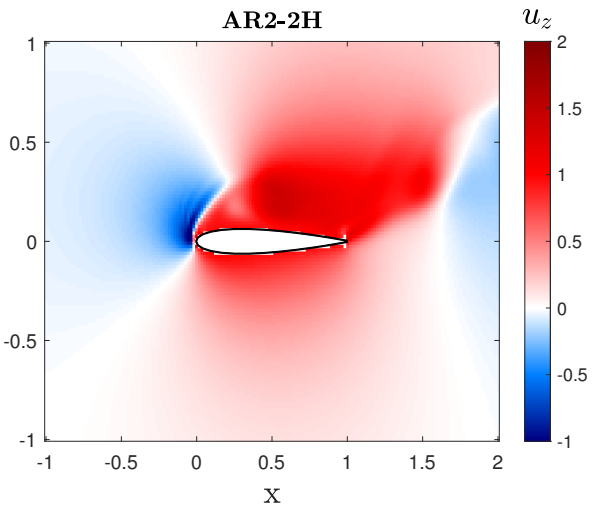


Figure 24: u_z field (50% spwise) ($\frac{t}{T} = 0.25$)

3.3 Transition from Flapping to Heaving

During this Section they will be compared different models simulating two symmetric wings with the same geometry but different kinematics. For this, variable R will be changed from $R = 0$ (Pure Flapping) to $R \to \infty$ (Pure Heaving) to see the influence of this parameter on the forces applied to the wing. The Aspect Ratio for all the wings will be fixed to 2.

3.3.1 Vortex Structures

In the next page it will be introduced the different vortex structures to see how this parameter varies with R. They are plotted at the midspan of the wing and for $\frac{t}{T} = 0.25$

As previously mentioned in Section 2.1, the velocity at the tip for any model will be the same due to the constraint imposed in Equation 1 of $h_0/c = 1$. This means that for the midspan of the wing, the velocity for each model will be smaller as R decreases.

Looking at vortex Figures 25, it could be seen that the LEV is stronger as the value of R increases. It is possible to see for the case of $R \to \infty$ and $R = 8$ that the LEV is detached, in the sense of experiencing a LEV relatively away from the wing surface. On the other hand, for R=0 case, it is shown a developing LEV which can be considered to be attached, in the sense of being closer to the wing surface. These differences will have direct consequences on the results of the forces.

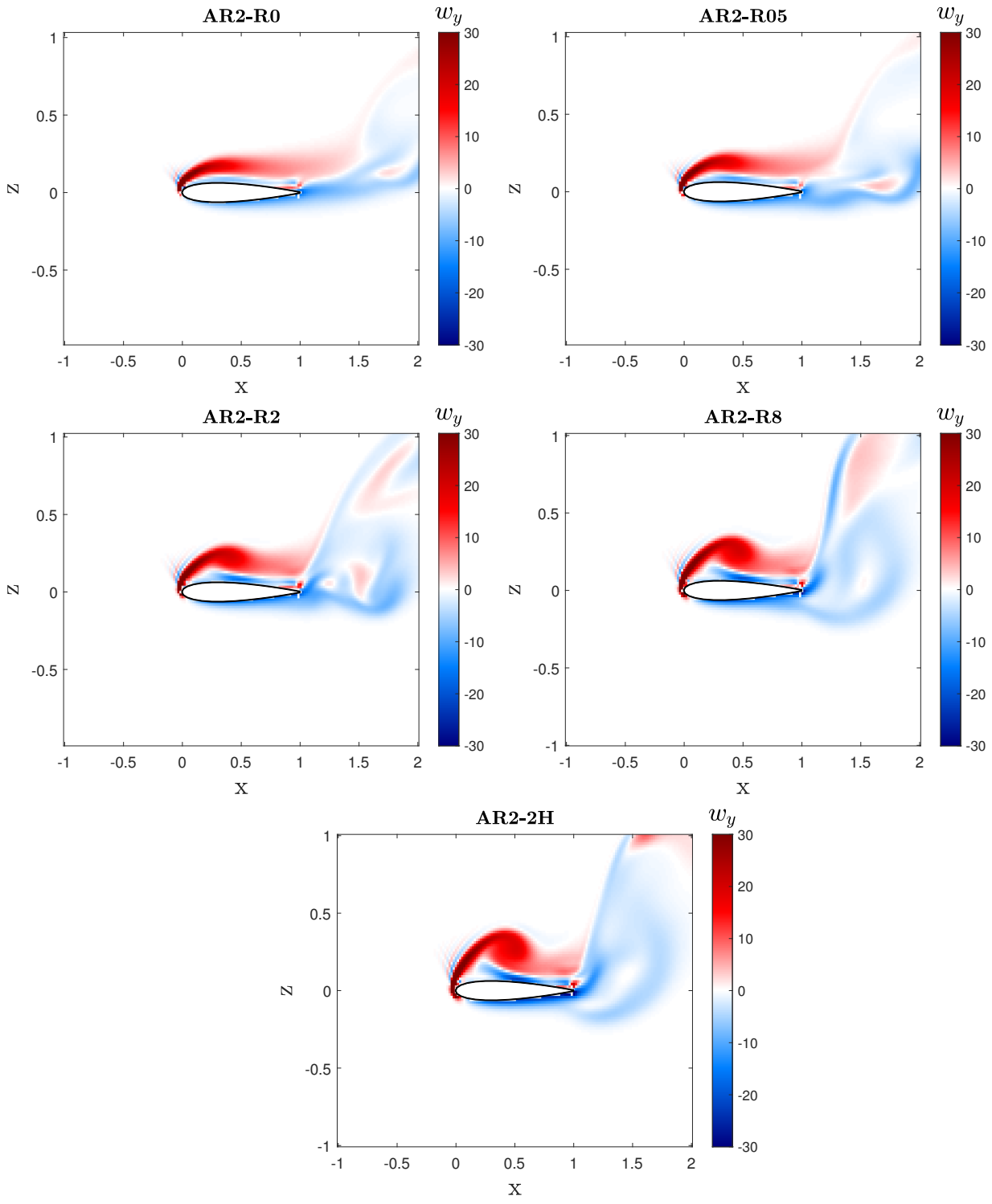


Figure 25: w_y for the transition from $R = 0$ (Heaving) to $R \to \infty$ (Flapping) Motion

3.3.2 Decomposed forces analysis: F_V and F_S

For seeing the difference between this transition, it has been plotted in Figure 26, (next page), the contribution of F_V and F_S, with the corresponding total force of each studied case. This was done to see the influence that each vorticity term has on the total force.

For C_D (left column in Figure 26), it can be observed that the F_V contribution will get more negative values (so that more thrust) when R grows. Furthermore, during the downstroke, the positive thrust will be reached before for those models with a bigger R. Attending to the F_S plot (Figure 26c), for all the cases it will be created positive drag, being the maximum peak at the middle of the downstroke for R=0. The direct consequence that this decomposition will have on the Total C_D force will be represented in Figure 26e, where it can be observed that it will be only created positive thrust for $R \to \infty$ and $R = 8$. So it can be affirmed that for the models proposed for the project, the body could only fly in forward flight for $R \to \infty$ (heaving) and $R = 8$.

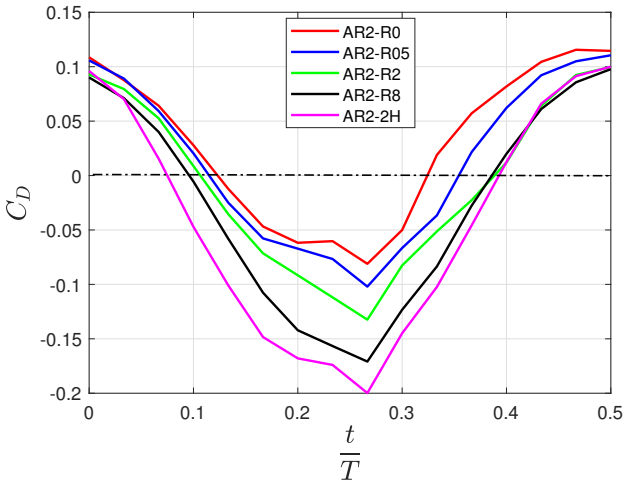
For C_L (right column in Figure 26), it can be seen that for $\frac{t}{T} = 0.25$, the positive contribution of F_V on C_L will be higher as R increases. Nevertheless, for the first stages of the motion it is checked that the lift contribution is smaller for high values of R. This is due to the higher velocity that the wing tip has for those models with small R during these first frames. As a consequence of this higher velocity at these first stages, it will be created a higher vortex intensity, so that higher F_V. Furthermore, there are even some stages for the highest values of R in which it is created negative lift. Attending to the F_S, it can be seen that the higher values of lift are created for models with high R.

For the resulting force in Figure 26f, it can be seen that for all the models the lift created will be positive except for the last stages of the downstroke where the wing starts to slow down.

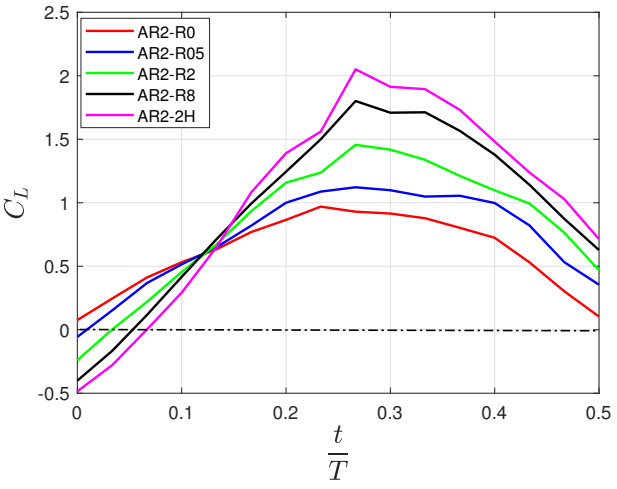
3.3.3 δ_t and δ_l analysis

Once the force analysis has been performed, during next two pages it has been presented the δ_t and δ_l maps for the 50% span for each wing when $\frac{t}{T} = 0.25$. They correspond with Figure 27 and 28 respectively.

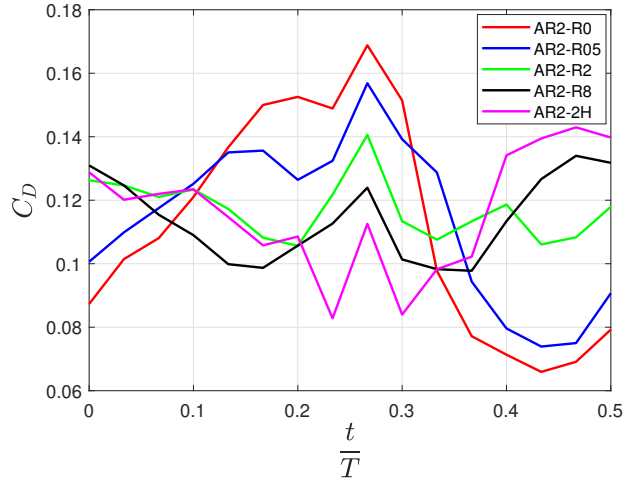
Through these plots it can be demonstrated the analysis previously introduced: for the models with small R, the LEV will be attached. As a consequence, the thrust and lift densities mapping will follow a similar shape, being the main force contributions attached to the wing surface. On the other hand, for models of higher R, it can be seen that the densities mapping are more detached.



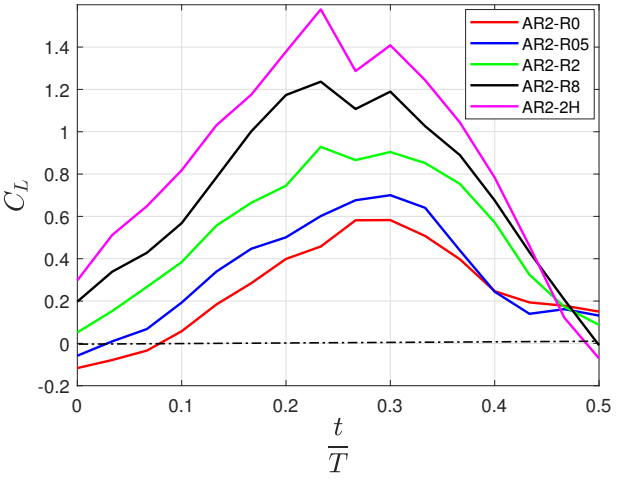
(a) C_D Vorticity within the flow - F_V



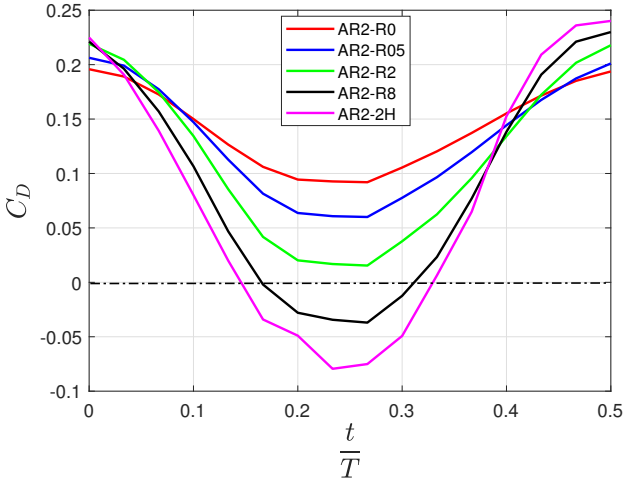
(b) C_L Vorticity within the flow - F_V



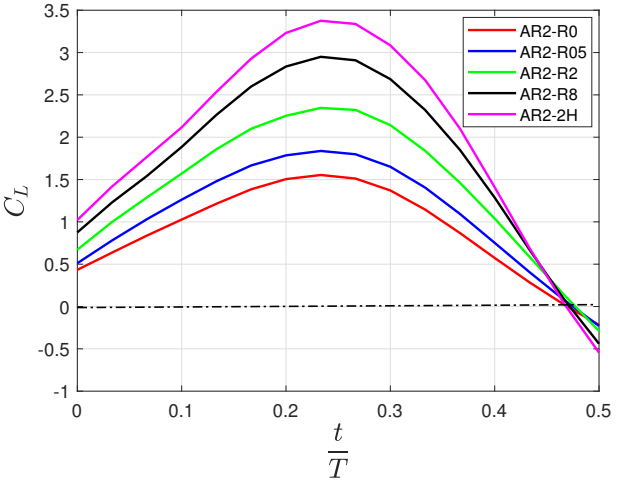
(c) C_D Surface vorticity - F_S



(d) C_L Surface Vorticity - F_S



(e) C_D Total



(f) C_L Total

Figure 26: Force Decomposition for different R and AR=2

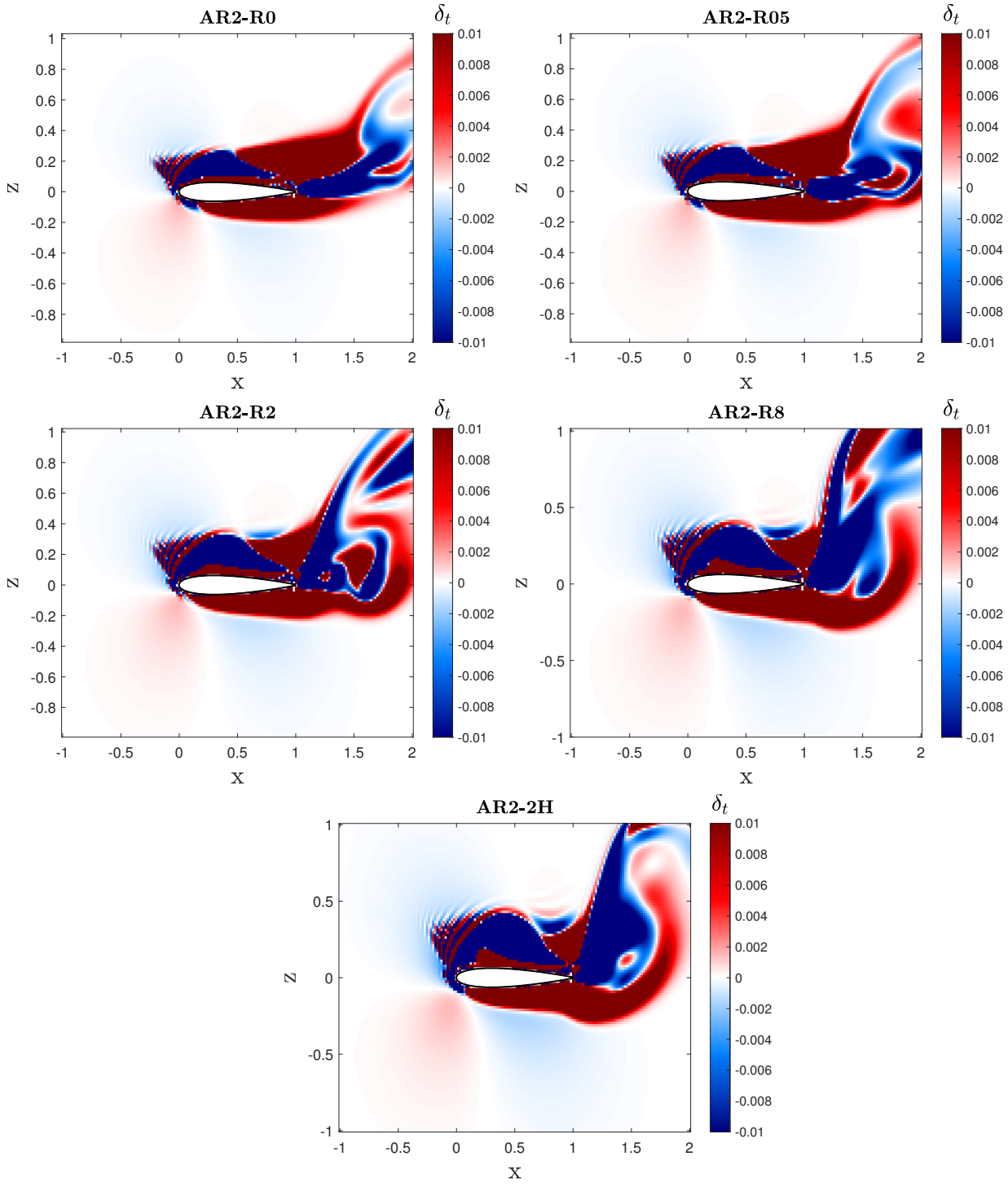


Figure 27: δ_t at the midspan for the transition from Heaving to Flapping Motion

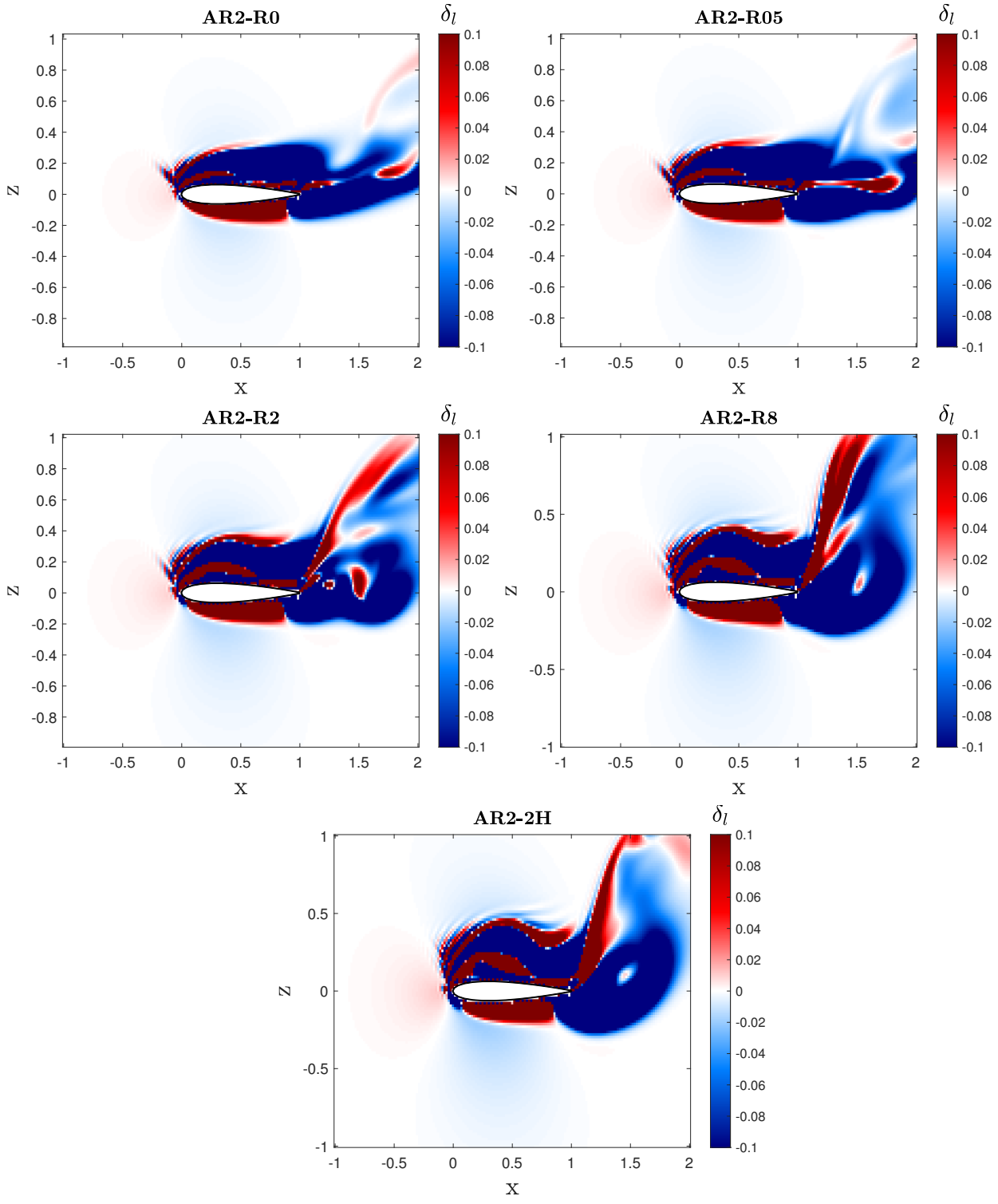


Figure 28: δ_l at the midspan for the transition from Heaving to Flapping Motion

3.3.4 SpanWise study for Heaving and Flapping

For better understanding of the results obtained for the decomposed forces at different R, it will be compared the two extreme values of R, so that flapping and heaving.
The thrust and lift densities obtained at the middle of the downstroke, will be plotted at different sections of the span: 20%, 40%, 60% and 80%. The inboard and outboard tip wing have not been studied for avoiding the presence of wing tip vortex.

With the analysis previously done, it could be affirmed that the LEV becomes less intense in the inboard side of the wing as R decreases. On the other hand, the vortex structure close the outboard wing tip, where the kinematics is almost the same, seems to have a similar behavior.

The discrepancies that could appear at 80% of the midspan are because the velocity for both cases will not be the same, being for the case $R \to \infty$ slightly bigger. For checking this, both vertical velocities at 80% midspan have been plotted.

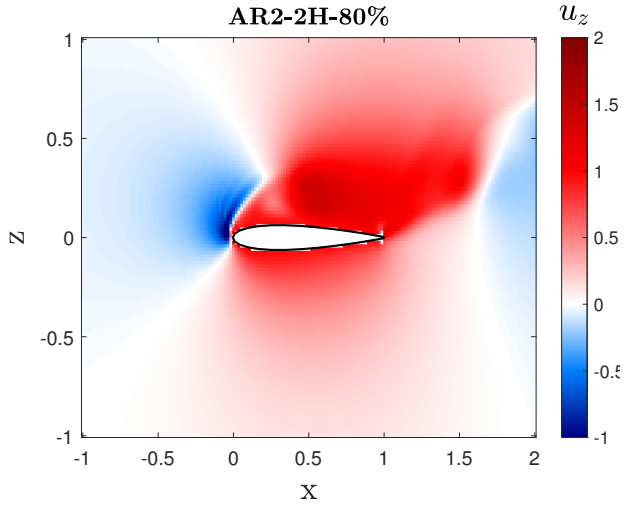


Figure 29: u_z field for $R \to \infty$ (80% spwise)

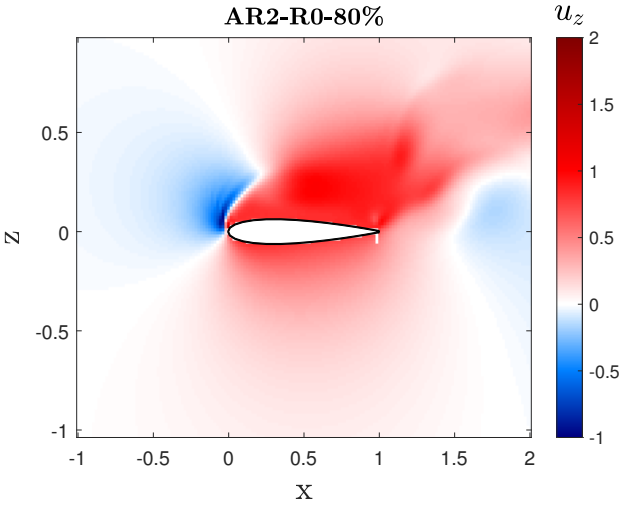


Figure 30: u_z field for R=0 (80% spwise)

This fact demonstrates the differences for the total force values between Heaving and Flapping motion. While in Heaving motion, the thrust and lift densities function follows a similar shape through the spanwise, for the flapping motion they change with the spanwise. This causes that for flapping motion there are zones on the wing which does not contribute enough to the lift and thrust generation, and as a consequence, forward flight is not possible.

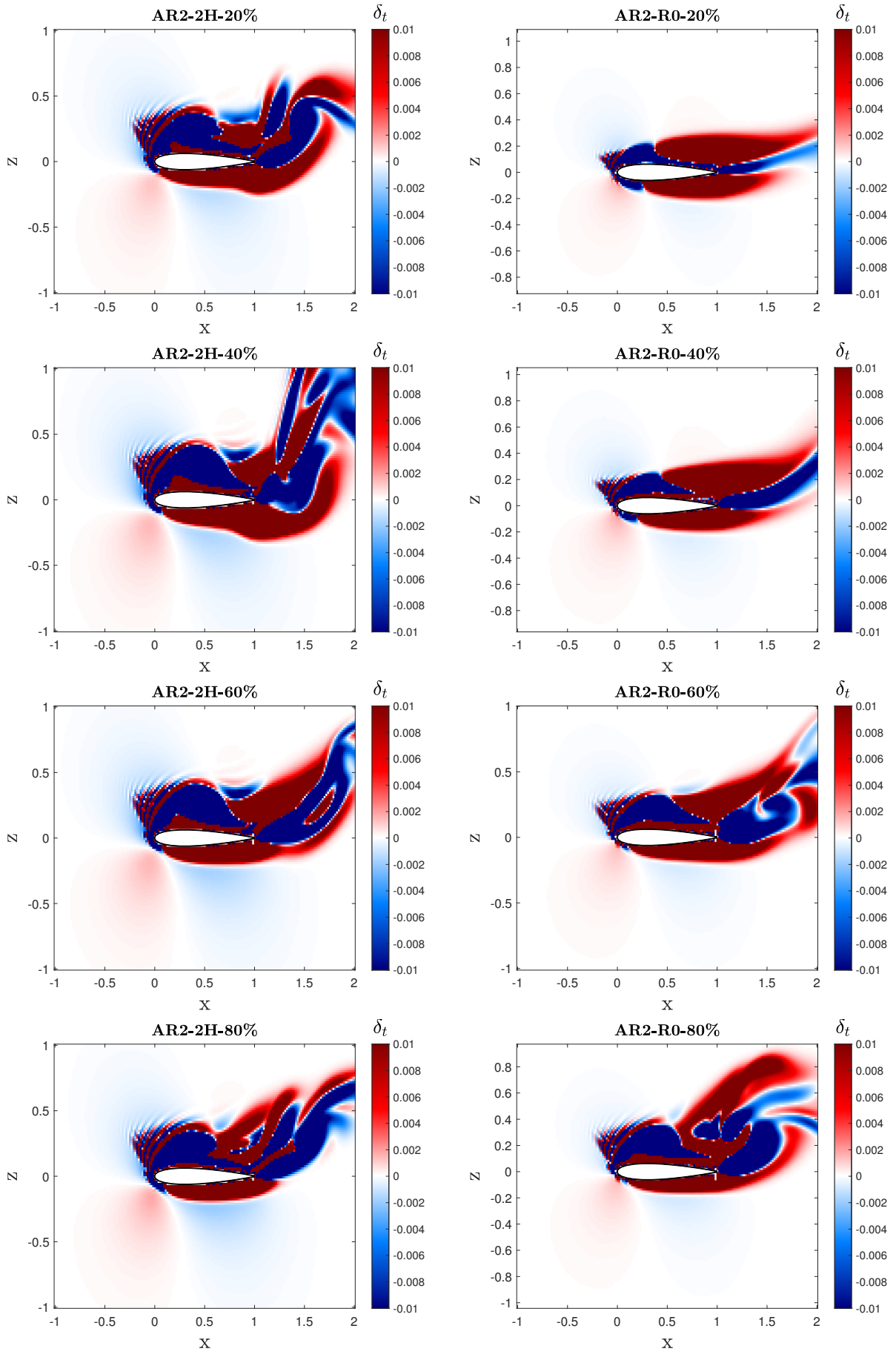


Figure 31: δ_t at different % of midspan for $R = 0$ and $R \to \infty$

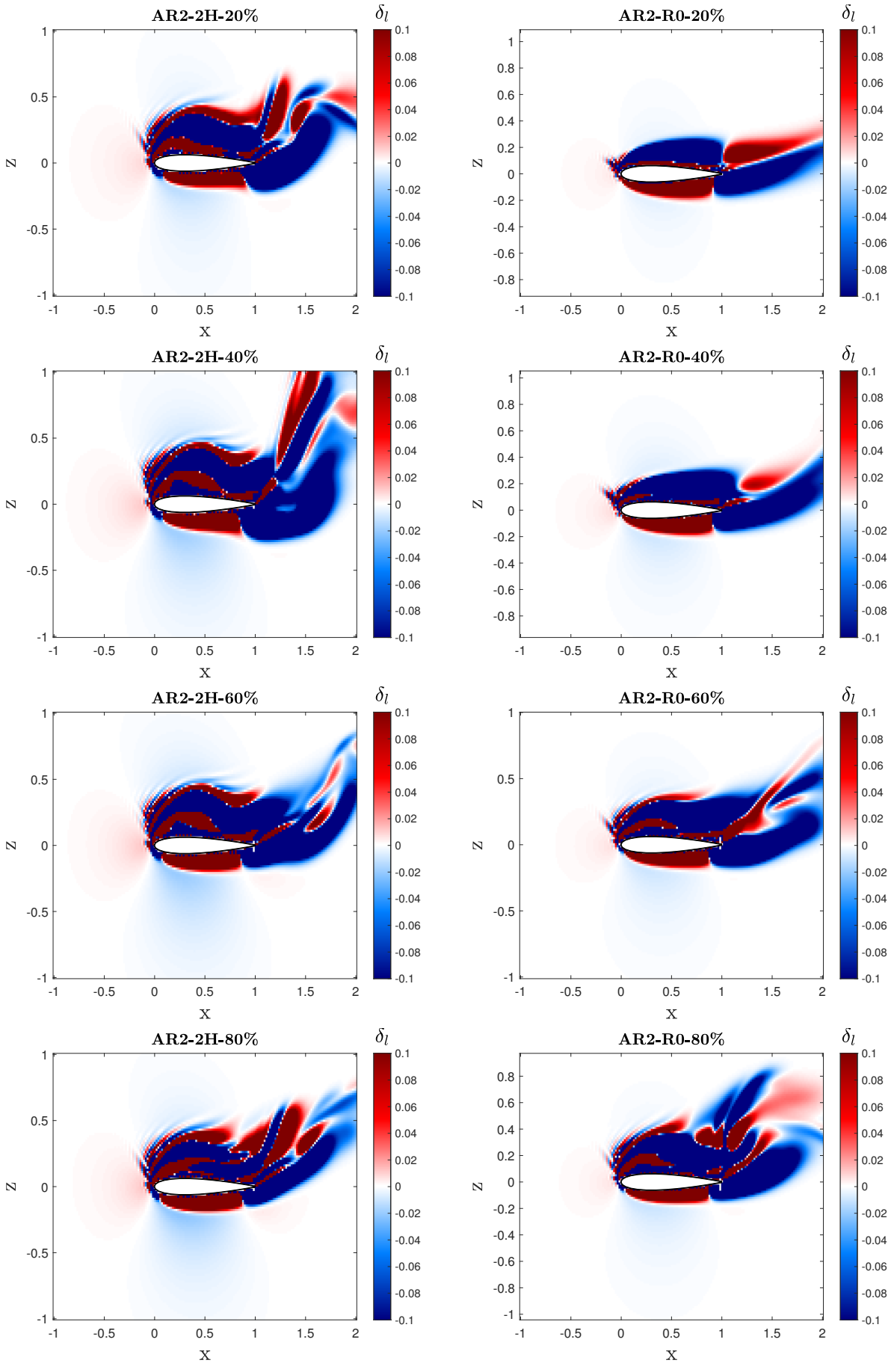


Figure 32: δ_l at different % of midspan for $R = 0$ and $R \to \infty$

3.4 Aspect Ratio Comparison: AR=2 and AR=4

Through this last Section, it has been compared two models which shares the same kinematics constraints but they have different geometric design. The kinematic imposed is a flapping motion with a Radius of Rotation of R=0 (Pure flapping). For the geometric part, an Aspect Ratio will be changed from 2 to 4.

In Figures 33 and 34 it is plotted the densities map at the middle of the span. It could be observed that this change in geometry, will not decisive variate the map solution close to the wing surface. This is again, due to the constraint of the velocity at the tips.

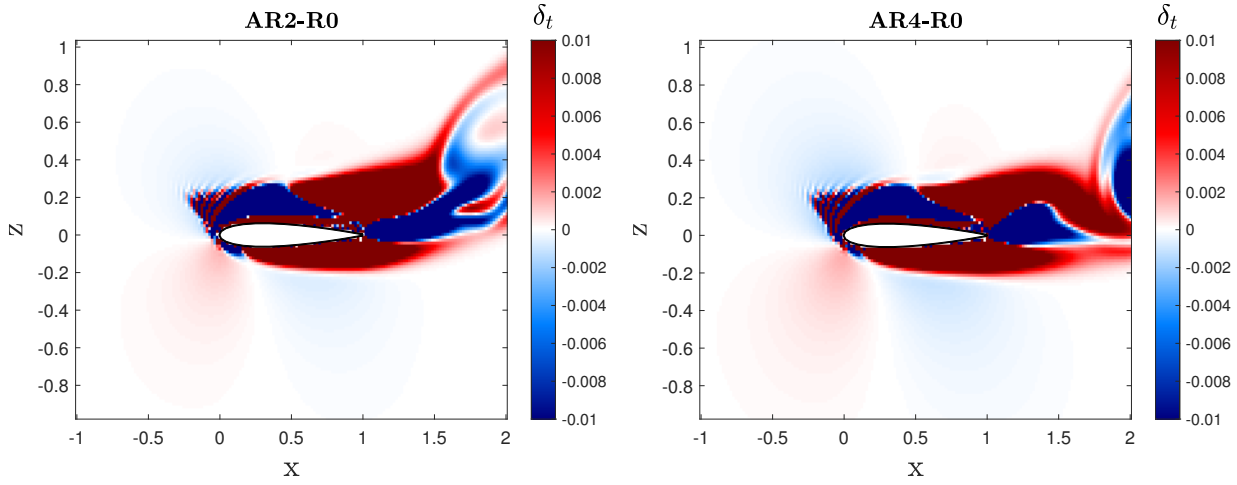


Figure 33: δ_t for AR=2 and AR=4 at the midspan when R=0

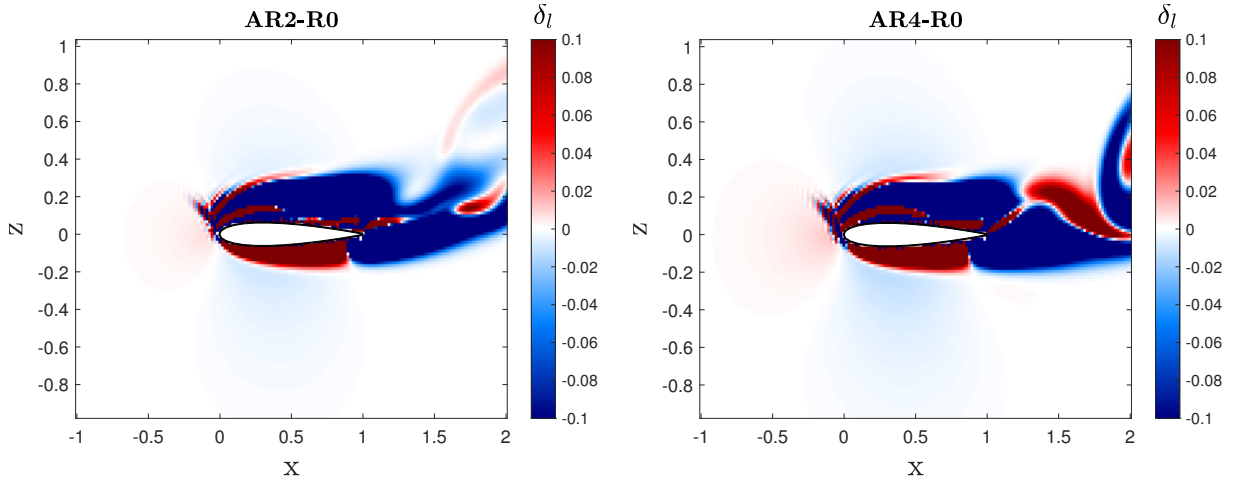


Figure 34: δ_l for AR=2 and AR=4 at the midspan when R=0

After this, it has been done a midspan comparison, where the same conclusion from the last section is reached: The kinematic constraint at the wing tip, makes that the results in the thrust and lift densities are the same for those sections which have the same vertical velocity. For this case, each % of midspan will have very similar density mapping.

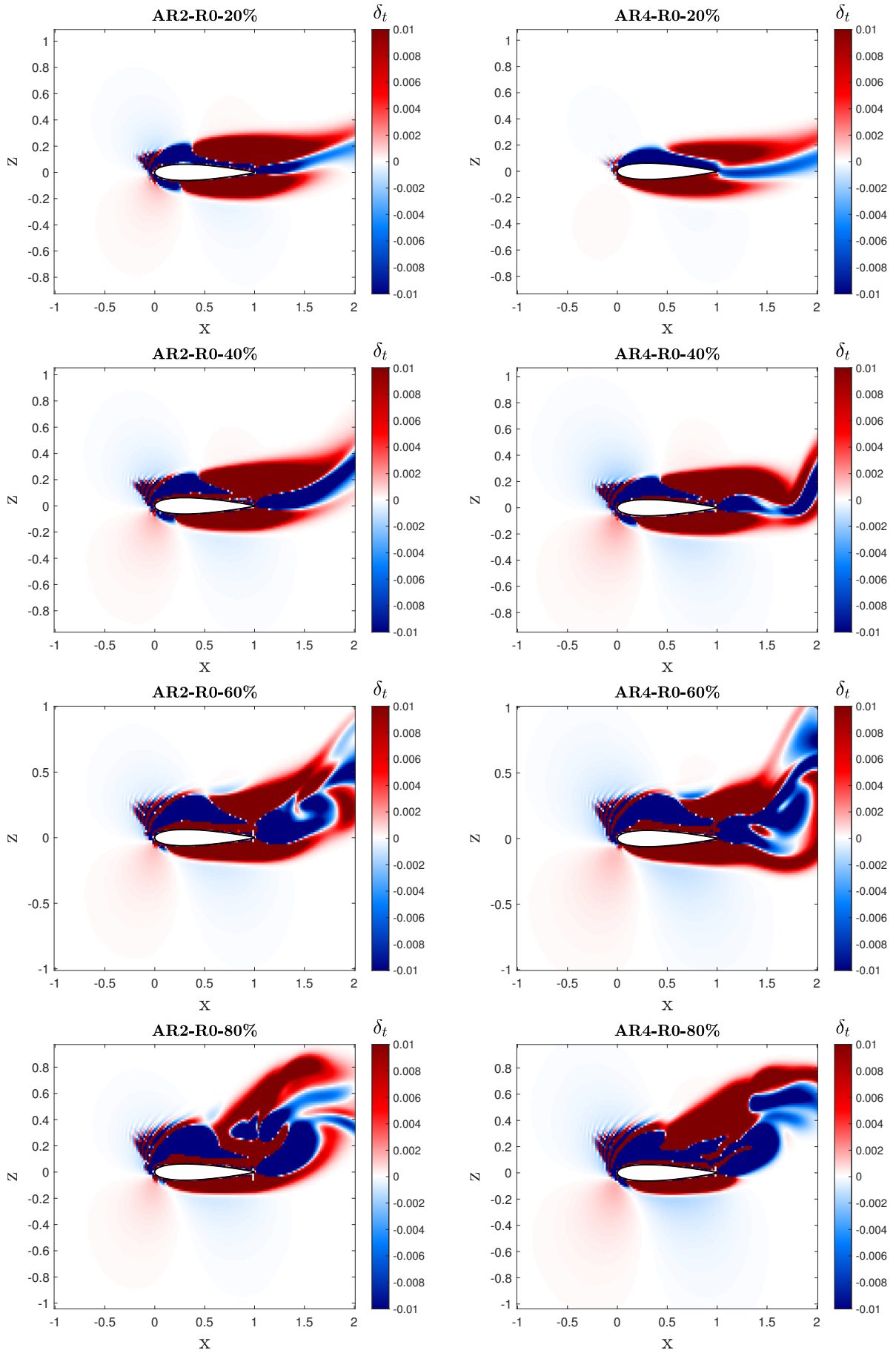


Figure 35: δ_t at different % of midspan when R=0. Left is AR=2 and Right AR=4. 46

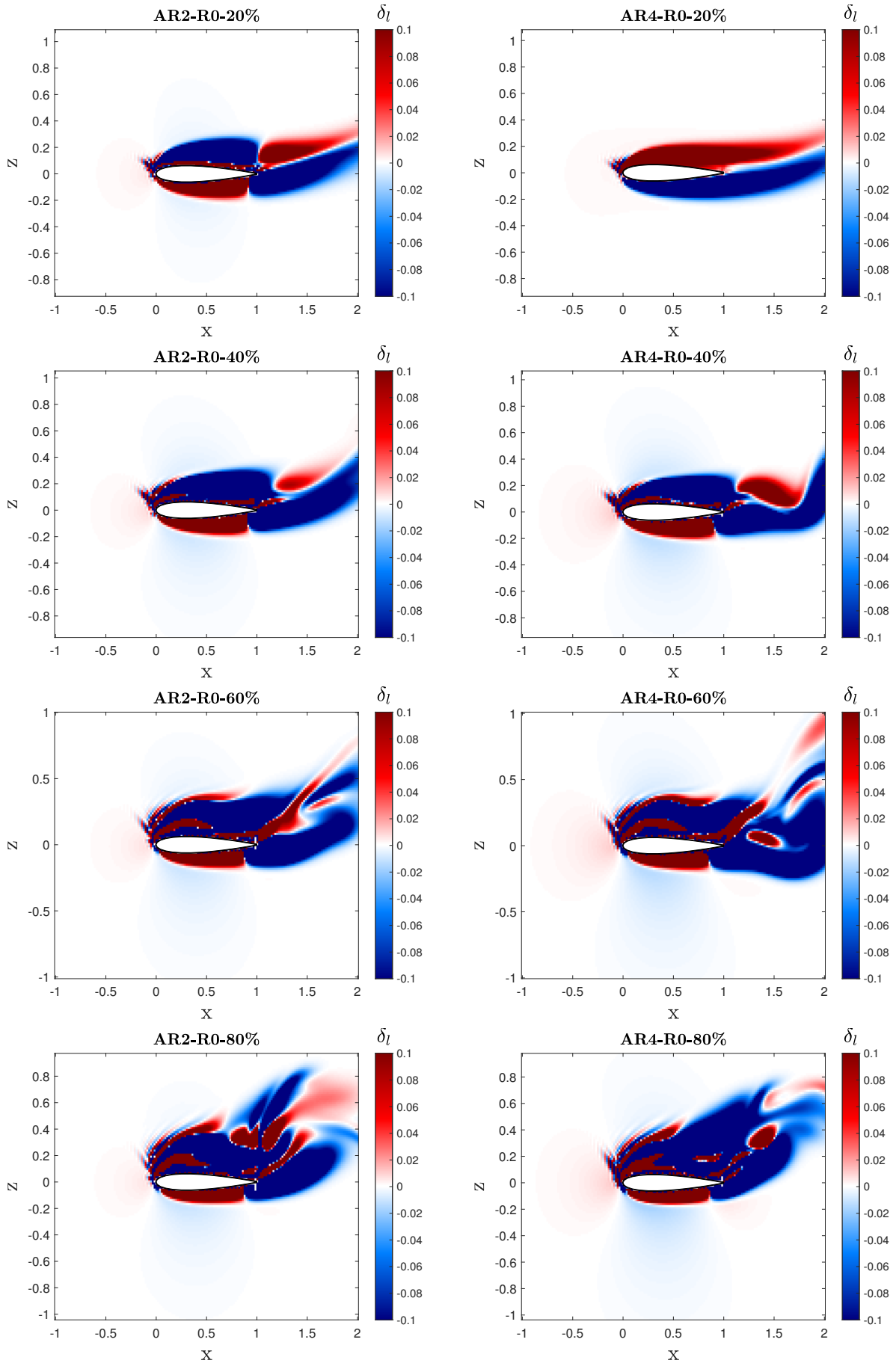


Figure 36: δ_l at different % of midspan when R=0. Left is AR=2 and Right AR=4.

4 Conclusions

During this project, the main goal was to implement a tool which is capable of computing the decomposition of forces for a pair of symmetric wings. This tool should be able to compute this information for different kinematics and geometric conditions. For this, a previous code where the only available input was 1 wing for heaving motion, has been modified.

Numerous tests have been performed for obtaining the most optimal way of computing the geometric and kinematics data of the second wing. The final decision was to compute this information for both wings with Matlab code and then input both wings data in the following programs. After that, it has been validated the BEMLIB libraries for a 3D body (sphere) comparing the potential flow results with the analytical ones.

Later, an error-time trade off analysis has been implemented for reducing the computational time as much as possible with the minimum error in BEMLIB potential flow computations. The solution gives 3926 points per wing with 1.7% average error. For the volume computations, it has been taken into account the control volume surrounding the right wing, but inputting the aerodynamic data of the left one.

Once the code is adapted to the new requirements, an extra process in which the computational time is reduced by half is implemented. This is related to the equation used for computing the potential gradient, where it appears a term for the distance from the wing to the CV points, which is proportional to the inverse of the fifth power. This causes that there are points situated a distance R_{Opt}, whose contribution is negligible. For this, an algorithm is created to detect these points and optimize the computational time. After this, another time-error trade off analysis has been done where $R_{Opt} = c$ seems to be the most optimal option with 0.05% average error (compared with no optimization process).

Moreover, different models are simulated with the new tool. It has been studied the influence on the forces that the second wing has on the first one. During the simulations, an inflow is set to $U_\infty = 1m/s$ with reduced frequency k=1 and Reynolds number Re=500. The conclusion is that the force results obtained do not differ decisively from one to two wing simulations. This is because from the DNS flow data used, there are not interferences between 2 wings in the spanwise direction.

After this is demonstrated, there have been studied and computed different models with $AR = 2$ (same geometry) where a transition from heaving to pure flapping (different kinematics) is simulated. The results say that the heaving motion is the one which produces the maximum thrust and lift for the vorticity within the flow and surface vorticity terms. In addition, forward flight is only possible for $R = 8$ and $R \to \infty$. This is due to the kinematics constraint imposed for all models at the wing tip where the vertical velocity is the same regardless the value for AR and R.

Finally, a comparison between two models with AR=2 and AR=4 (different geometry)

and both in pure flapping (same kinematics) has been done. The conclusions are the same than when the kinematics were changed: the kinematic constraint at the tip, makes that when the geometry is changed, the results do not vary between each other.

With all this, it can be concluded that the tool has been successfully built. In addition, through this new tool, there have been studied different cases for which the results coincide with recent studies done by Carlos III University on behalf of Alejandro Gonzalo in his doctoral thesis [14].

5 Future Work

As the big flapping vehicles have been demonstrated to be inefficient, the future work for flapping industry will be focused on MAVs. Here are included flapping devices which fly at a low Reynold number.

With the code configured and perfectly built, it is significant to analyze the future work that can built on top of it. This Section is in charge of explaining possible work that could be continued by the tools and results figured out during this project.

So following the scope treated in this project, the simulations performed can be done with denser meshes to fully affirm the obtained results. The ideal wing refinement mesh could be to obtain 56 points per chord instead of 15. This new refinement would correspond with one used for the DNS data. For this, more optimization process should be performed, specially to the BEMLIB libraries to reduce as much as possible the computational time of the potential flow. At the same time, this should avoid memory performance overhead to allow the implementation of denser meshes.

If these refinement and optimization steps are achieved, the next action could be to impose new geometric and kinematic conditions for the wing analyzed. For this, it could be input non-zero values to Φ_m and φ in Equation 1. Another study that could be done would be including the pitching motion which is present in all the small insects. These new models can build on top of other studies, as the one performed by Moriche in [6], where the decomposition of forces is done for these motions but from a 2D point of view.

All these future steps, will help to increase the current knowledge about flapping systems. They will allow to predict the forces which are in charge of certain flights allowing the implementation of control systems in MAVs.

6 Budget

Through this section it will be presented the needed budget for developing the project including the researcher time, the computer software licenses needed employed and the hardware systems.

Nowadays, the average salary for a junior researcher in Spain is 10€/hour. The estimated time by Universidad Carlos III for doing the project is 360 hours which makes a total crew salary of 3600 €.

Looking at the software needed, *gmsh* and Fortran are free. On the other hand, despite the Matlab license is free for students, the University is in charge of this spend with a total value of 500€. The BEMLIB libraries and all documents and websites used during the project were free because most of them were taken from Universidad Carlos III library.

During the computation of the code, and concretely the part of the potential gradient where a high amount of points are treated, a node in a cluster was rented. This hardware is composed by 12 cores and the approximate total price is about 2.4€/hour. Since the estimated total computational time is 2000 hours, this makes a total cost of 4800 €.

Summarizing all these spends, the total cost of the project is about 9000 € with a total time needed for performing this work of 6 months.

7 Socioeconomic impact of the project

For manufacturing a big flapping wing, the high inertial forces applied to this part deals in strict structural requirements which at the same time, induces in difficulties for obtaining a wing enough light. These technical requirements are under a commercial requisite where the construction of these structures seems to be very expensive and not profitable.

Nevertheless, since the flapping system has been demonstrated to be only efficient for low Reynold Numbers, MAVs can be only compared with drone sector. This is why small flapping systems can start competing on Drone Market where there exist micro devices that operate with small propellers. In this drone sector, each milligram counts so this is a very important constraint for heavy onboard systems, specially the batteries. The lightness of these devices, lead to increase the fragility for stresses and crashes. So a quick aerodynamic response should be done for avoiding obstacles and changes in environment. Takig into account this, studies have demonstrated that flapping wings overcome fixed wings [36]. The best example for this is the high maneuvility that hummingbirds have.

So looking at the drone market, where flapping devices would operate, it is expected that this sector acquires a value of $100M in 2020. From these predictions, 70% of this market share will be destined to military purposes where four out of five top military UAV manufactures are American [39]. Then, 17 % of the share will be the consumer market where the Chinese giant DJI dominates this sector with over 70% for this market share. Parallel, the industrial sector will appreciate a considerable increase of the market set, being the main applications in mapping, surveying, maintenance and inspection. Gradually, the drone market is also introduced in the agriculture sector by creating drones capable of autonomously spraying, planting seeing or doing satellite imagery.

During last years, the agriculture sector has demanded a device capable of pollinate flowers for increasing or maintaining the production of certain aliments as the honey. MAVs with flapping system which simulate honey bees can be the best solution for this issue. This needed is related with the population decline of this animal in recent years [37]. More opportunities for these devices can be on the commercial share where lately leisure and micro-vehicles are closely linked.

All the current MAVs which have been designed to this day, are created under aerodynamic parameters which are not full-fill understood. The main objective for this project was to advance as much as possible the actual knowledge of the forces which are involved in the flapping wings. With this, the flight efficiency can be improved with the consequence of less power consumption, which means less battery weight. This increase in the efficiency of the devices could deal in an overall increment of the revenues for the company in charge of promoting flapping devices. So the MAVs market can quick enhance if the companies start developing flapping systems. For doing this, sufficient theoretical arguments are necessary for the construction of these air vechicles.

8 Regulatory framework

In Spain, the actual legal framework related with MAVs is the same as any propeller drone in the Decreto1036/2017[2] which is available in BOE from December 29^{th} 2017. Here it appears the liabilities and responsibilities for the pilot and the required geometric and performance conditions for the air vehicle in each situation. [38]

From Article 21.3, for devices which do not require navigability certification, they can be used in population places under the following conditions: Maximum Take of Weight (MTOW) of 10kg, Visual Line of Sight (VLOS) with a maximum value of 100 m, a maximum height from the ground of 400 ft (120 m) or a distance from the highest obstacle of 600m from the aircraft. Any operation should be done in no air controlled areas and out from the Flight Information Zone (FIZ).

On the other hand, for purposes which require aircraft with navigability certification, the limitations imposed for the flights will depend on the certification imposed by Agencia Estatal de Seguridad Aérea [3] for each pilot and device.

For the experimental flights, which correspond with most of flapping flights, from Article 23, those flights can be only done inside the visual conditions, in a delimited air space and outside from the population or inhabited zones. These flights should avoid closeness to outdoor meetings and they require a well-defined security space with no navigability control and out from FIZ.

During legal document it is treated others facts as the pilot requirements, meteorology, security, insurance or countries bounderlines. MAVs are a recent technolody so there is no information or specifications about these particular systems. This market needs to launch a decisive model to start the production and a specified regulatory framework will be needed for flapping systems.

[2]BOE Decreto 1036/2017
[3]AESA

References

[1] Gottfried Sachs. Comparison of Power Requirements: Flapping vs. Fixed Wing Vehicles. Institute of Flight System Dynamics, Technische Universität München, Boltzmannstr. 15, 85748 Garching, 2016

[2] Dario Floreano and Robert J Wood. Science, technology and the future of small autonomous drones. Nature, 2015

[3] William Thielicke. TheFlappingFlightofBirds AnalysisandApplication. PhD thesis, University of Groningen, 2014.

[4] J Borrell, Brendan Goldbogen, Jeremy Dudley, Robert. Aquatic wing flapping at low Reynolds numbers: Swimming kinematics of the Antarctic pteropod, Clione antarctica. The Journal of experimental biology, 208. 2939-49. 10.1242/jeb.01733, 2005.

[5] Ellington, Charles van den Berg, Coen Willmott, Sandy Thomas, Adrian. Leading-edge vortices in insect flight. Nature. 384. 626-630. 10.1038/384626a0, 1996

[6] M.Moriche. A numerical study on the aerodynamic forces and the wakes tability of apping ight at low Reynolds number. 2017.

[7] W. Shyy, H. Aono, C. K. Kang, and H. Liu. An introduction to flapping wing aerodynamics. Cambridge University Press, 2013.

[8] Richard Peters. How to photograph birds. Photography Tutorials - Wildlife, 2014

[9] Song, Jialei, Tobalske, Bret W., Powers, Donald R., Hedrick, Tyson L., and Luo, Haoxiang. "Data From: Three-dimensional Simulation for Fast Forward Flight of a Calliope Hummingbird." 2016.

[10] Dececchi, T. Alexander. "Data From: The Wings before the Bird: An Evaluation of Flapping-based Locomotory Hypotheses in Bird Antecedents." 2016.

[11] A.Martín-Alcántara, R.Fernandez-Feria, and E.Sanmiguel-Rojas. Vortex ow structures and interactions for the optimum thrust eciency of a heaving airfoil at dierent mean angles of attack. Phys. Fluids, 27(7):073602, 2015.

[12] Hoang Vu Phan, Quoc Viet Nguyen, Quang Tri Truong, Tien Van Truong, Hoon Cheol Park, Nam Seo Goo, Doyoung Byun, Min Jun Kim, Stable Vertical Takeoff of an Insect-Mimicking Flapping-Wing System Without Guide Implementing Inherent Pitching Stability, Journal of Bionic Engineering, Volume 9, Issue 4, 2012

[13] Jong-Seob Han, Ho-Young Kim, and Jae-Hung Han Interactions of the wakes of two flapping wings in hover. Physics of Fluids 31, 021901 (2019)

[14] Alejandro Gonzalo Grande. Aerodynamic forces and vortex structures of flapping wings in forward flight. 2018

[15] Richard J. Bomphrey, Toshiyuki Nakata, Nathan Phillips, and Simon M. Walker. Smart wing rotation and trailing-edge vortices enable high frequency mosquito ight. Nature, 2017.

[16] Z.J.Wang. Vortex shedding and frequenc selection in flapping ight. J. Fluid Mech., 410:323–341, 2000.

[17] G. C. Lewin and H. Haj-Hariri. Modelling thrust generation of a two dimensional heaving airfoil in a viscous ow. J. Fluid Mech., 492:339–362, 2003.

[18] M. A. Ashraf, Young J., and J. C. S. Lai. Reynolds number, thickness and camber eects on apping airfoil propulsion. J. Fluids Struct., 27(2):145–160, 2011

[19] G.C.H.E. de Croon, M. Percin, B.D.W. Remes, R. Ruijsink, and C. De Wagter. The DelFly design, aerodynamics, and articial intelligence

[20] Zhi Ern Teoh, Sawyer B Fuller, Pakpong Chirarattananon, NO Prez-Arancibia, Jack D Greenberg, and Robert J Wood. A hovering apping-wing microrobot with altitude control and passive upright stability. In Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.

[21] Hoang Vu Phan, Taesam Kang, and Hoon Cheof Park. Design and stable ight of a 21 g insect like tailless apping wing micro air vehicle with angular rates feedback control. Bioinspiration Biomimetics, 2017.

[22] C.-C.Chang. Potential ow and forces for incompressible viscous ow. InProc. R. Soc. Lond. A, volume 437, pages 517–525, 1992.

[23] Carlos Paulete. Force decomposition in 3D unsteady aerodynamics, Universidad Carlos III, 2018

[24] Weinberg Gabriel DuckDuckgo. Quora. https://www.quora.com/How-is-lift-generated-with-symmetrical-wings (accessed March 5, 2019)

[25] Airfoil database search. www.airfoiltools.com/search/index?m[grp]=naca4dm[sort]=1 (accessed March 30, 2019)

[26] M. Uhlmann. An immersed boundary method with direct forcing for the simulation of particulate flows. J. Comput. Phys., 209(2):448–476, 2005.

[27] C.Pozrikidis. A Practical Guide to Boundary Element Methods with the Software Library BEMLIB. Chapman Hall CRC, 2002

[28] A three dimensional finite element mesh generator with built in pre and post processing facilities http://gmsh.info/ (accessed April 3, 2019)

[29] Tzvetalin Simeonov . Optimal Area Triangulation. University of Saskatchewan Saskatoon, 2005

[30] N. P. Weatherill. Delauny Triangulation in Computational Fluid Dynamics. Institute for Numerical Methods in Engineering University College of Swansea Singleton Park, Swansea, U.K, 1992

[31] W.H. Press, Teukolsky S.A., Vetterling W.T., and B.P. Flannery. Numerical Recipes in Fortran 77: The Art of Scientific Computing. Cambridge University Press, 1992.

[32] L. Quartapelle and M Napolitano. Force and moment in incompressible flows. AIAA J., 21(6):911–913, 1983.

[33] David Dahl (Idaho National) SIMION: Unique Theorem. Scientific Instrument Services, Inc. (SIS), 2016

[34] https://foxnomad.com/2018/05/03/how-much-does-a-plane-turn-on-its-side-when-turning-in-the-air/

[35] WolframMathWorld, Kronecker Delta. http://mathworld.wolfram.com/KroneckerDelta.html (accessed May 5, 2019)

[36] S. N. Fry, R. Sayaman, and M. H. Dickinson. The aerodynamics of free-flight maneuvers in drosophila. Science, 2003

[37] Scientists develop flying "robo-bees" to pollinate flowers as bee populations decline. https://inhabitat.com/scientists-develop-flying-robobees-to-pollinate-flowers-as-bee-populations-decline/. (accessed June 9, 2019)

[38] Agencia Estatal Bolitín del Estado. https://www.boe.es/ (accessed June 5, 2019)

[39] Drone Enthusiast. Commercial Drone Market. Commercial Done Market. https://www.dronethusiast.com/commercial-drone-market/ (accessed June 5, 2019)